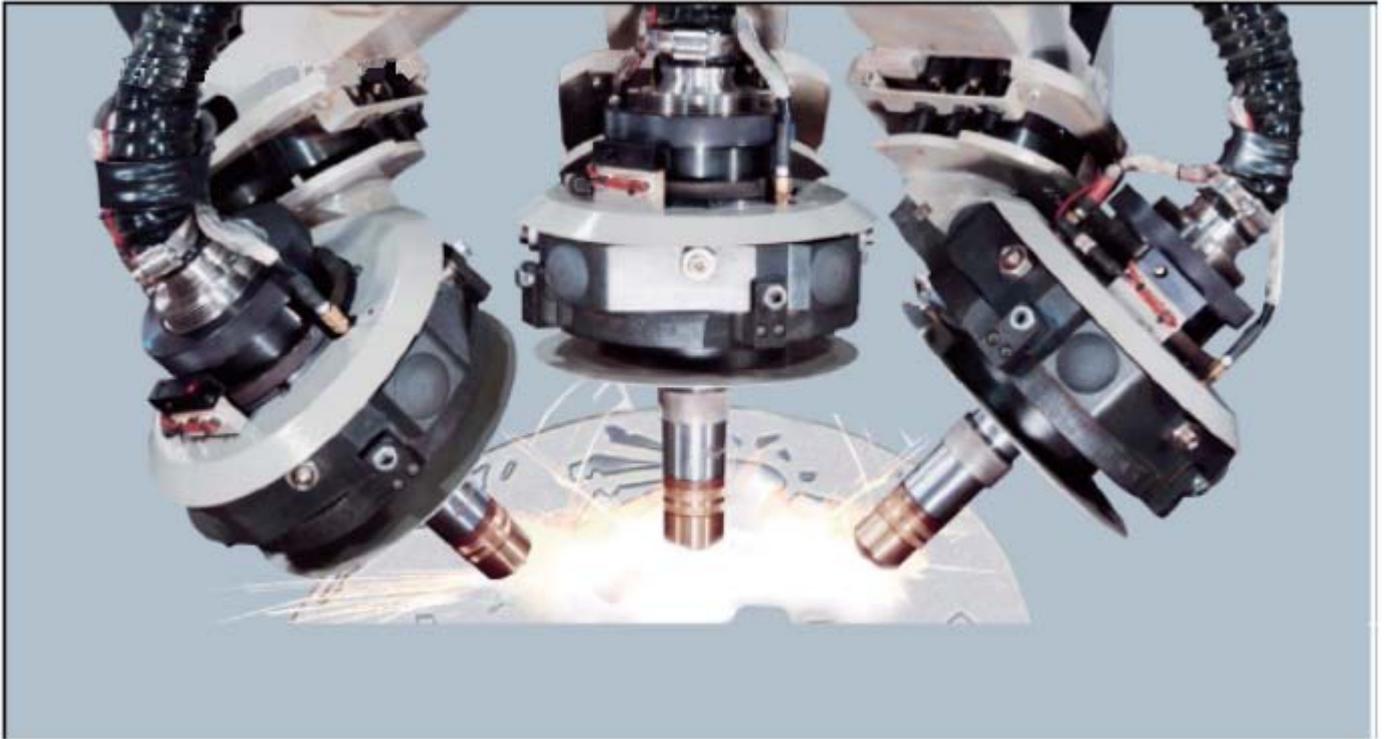


BECKHOFF 自动化新技术



TwinCAT CNC 简明调试教程

Version 2.0

毕孚自动化设备贸易（上海）有限公司

系统应用部 史晓云

2014 年 05 月

本文纯属个人经验，非 Beckhoff 公司正式放行，如有疑问，请直接联系作者

前 言

随着科学技术的发展，制造技术有了深刻的变化。在制造领域，尤其是金属加工领域，数控技术已经带来了革命性的影响。传统的 NC 系统使用固定的逻辑单元控制程序，不能够由程序员或者机床操作人员更改，可以称为硬连接，新的数控技术 CNC 系统可以由程序员通过控制系统进行程序修改，这突破了传统机床对于高效率、高质量、多样化的发展桎梏，数控技术对于制造业也影响逐渐扩大。随着计算机计算的发展，数控技术也得到了一次新的革命，从画面显示单元、逻辑控制单元、运动控制单元各自独立的控制架构集成在以微软操作系统为平台的一体式控制。提高了计算机的 CPU 利用率，更加灵活、便捷的实现画面修改、运动分解等工作。

以倍福为代表的软控制系统，基于微软操作系统的控制架构，具有了灵活、易操作的基础优势。从简单地逻辑控制到复杂的 CNC 数控技术都秉承了这样的天赋。以机床发展为例，PLC 继电器控制、点位控制的 NC PTP、简单凸轮齿轮的 NC PTP、中型数控技术的 NCI（三轴插补，五轴联动）、大型的数控技术 CNC 系统。

大型的数控技术 CNC 系统是金属加工行业的技术平台，是一套标准的 CNC 系统，本身 CNC 代码遵循 DIN66025 的标准。可以实现多轴插补、空间变换、坐标机械人、DELTA 机器人、关机机械臂等复杂的空间轨迹变换。基于开放的 windows 平台，人性化的界面，组成了有别于基于硬件的软 CNC 系统。基于倍福 CNC 系统的开发的机床系统包括铣床、车床、冲床、激光切割机床、火焰切割机床、水切割机床、磨床、剪板机床以及各种木工机械等非金属加工机床等。

倍福 CNC 系统的学习包含两部分：TWinCAT PLC 程序的学习，即逻辑控制部分的学习；CNC 系统部分学习，包含 G 代码学习以及 CNC 与 PLC 接口部分学习。有条件建议部分高级语言 Visual Studio C# 学习。

本手册纯属个人知识总结，内容难免存在错误或者不妥之处，如有错误，恳请批评指正。



0.1 本书读者对象

本书的目的是帮助您快速掌握以 TwinCAT 为基础的软 CNC 运动控制系统的使用。对于 CNC 系统的使用需要您掌握 TwinCAT PLC 以及 TwinCAT NC PTP 的编程应用，并具备 CNC 系统的基础知识。本书在附录表中对于机床系统专业术语做了部分总结，便于学习。

本书适于以下情况：

- 对于 TwinCAT CNC 系统的初学者。

TwinCAT CNC 系统的初学者，CNC 系统涵盖了 TwinCAT 系统中的 PLC、NC PTP、CNC 系统，可以全面的学习运动控制涉及的从简单逻辑控制到初级点到点控制到复杂的插补算法、空间轨迹算法的完整的学习。TwinCAT CNC 适用于大专院校的学生和暂无实际项目需求的工程师以及实际项目的仿真模拟。TwinCAT CNC 模拟运行调试无需硬件，可以在 TwinCAT CNC 平台直接实现。CNC 软件的获得需要与供应商沟通获得。另一方面，TwinCAT CNC 系统遵循标准的 DIN66025 的指令语法结构，并具有更加灵活的辅助语法，因此，通过 TwinCAT CNC 学习高端运动控制，即可获得标准 CNC 的语法知识，更能够了解软 CNC 系统的灵活方便的特性。

- 对于曾经使用其它 CNC 控制系统，但第一次使用 TwinCAT CNC 的初学者。

CNC 控制系统常用功能，典型的控制语法都在本书中讲解。本书将学习最基本的 TwinCAT CNC 任务创建、控制轴和通道的配置、基本的框架界面，由浅入深实现 TwinCAT CNC 控制程序的编写、调试。

- 曾经使用 TwinCAT CNC 及其它运动控制系统的有经验的工程师。

CNC 配置方面，对于 TwinCAT CNC 系统的轴、通道参数进行了讲解、并对 PLC 控制部分的结构框架进行了讲解。

CNC 代码方面，对于常用的 G 代码如 G00\G01\G02\G74 等代码功能使用以及 MHT、VA、VG、VE 变量进行了说明；

界面部分，对于 HMIPRO 基础框架进行了分功能讲解，便于在实际工程中的使用。



0.2 本书的主要内容

第一章，TwinCAT CNC 系统概述；

第二章，TwinCAT CNC 系统软件安装操作；

第三章，TwinCAT CNC 系统编程指令；

第四章，TwinCAT CNC 配置文件；

第五章，TwinCAT CNC 工程 PLC 基础程序；

第六章，TwinCAT CNC 工程特殊功能介绍；

附录：TwinCAT CNC 界面软件及报警配置、机床专业术语等；

0.3 版本说明

本书提供的操作截图、程序代码都基于 TwinCAT 2.0。

关于 TwinCAT 3.0 部分的 CNC 介绍会略作介绍。

0.4 勘误表

尽管我们竭尽所能来确保在正文和代码中没有错误，但人无完人，错误难免发生。如果您在本书中发现了错误（例如拼写错误或者代码错误或者功能错误），我们将非常感谢您的反馈。发送勘误表将节省其他读者的时间，同时也会帮助我们提供更高质量的信息。

请发送邮件至x.shi@beckhoff.com.cn，该邮箱由作者本人查收，我会检查您的反馈信息。

如果正确的，将在本书的后续版本中使用。

0.5 感谢

用户的需求是我们成长的动力，在此感谢长期给予我们动力的客户。

本书的编辑和校对得到了 BECKHOFF 中国系统应用部部门同事的大力支持，在此特别感谢系统应用部经理王建成，我的同事王君、高昇、况云龙、黎凌霄、王兰立、刘记忠以及德国同事 Henning Rausch，老同事包俊杰等的无私帮助，感谢大家的经验分享以及辛勤工作。

最后，希望更多的人使用 TwinCAT CNC，做出更先进的机床，服务客户，为社会进步做出贡献！

第一章 TwinCAT CNC 系统概述

概述:

倍福 CNC 系统是建立在倍福的 TwinCAT NCI 软件基础上的一个应用。倍福 CNC 系统是一套标准的应用系统：G 代码格式遵循德国 DIN66025 标准，PLC 程序遵循 IEC61131-3 标准，配置界面在 C#程序下编写的。倍福 CNC 系统是一套完整的 CNC 应用系统，功能涵盖了普通的 CNC 插补功能，到复杂的空间坐标转换功能，RTCP\TLC\KINEMIC 坐标转换，可以实现多通道、多种类型 CNC 轴、高速切割、样条等标准 CNC 功能等；倍福 CNC 系统是一个完善的开发平台，适合用于磨床、钻床、铣床、车床、特殊机械如等离子、激光机床等。

一、控制硬件要求

工控机选择：建议采用 Intel Celeron 1.9GHz 2 core 处理器，2GB DDR2RAM 内存及以上；
操作系统：Windows 2000 / Windows XP / Windows 7。

二、控制软件

基础 TWinCAT 软件：TwinCAT 软件安装级别为 NCI；
CNC 安装软件：1506E、1506 版本或者 2010E、2010 版本等；
HMIPRO 界面软件：Windows framework 版本 2.0 及以上；

三、CNC 软件介绍

CNC 软件划分：CNC-E 版本、CNC 全版本、PACKAGE 软件

CNC-E 版本：即 EXPORT，默认版本，该版本支持：

- 1 个 CNC 插补通道；
- 4 根 CNC 插补轴；
- 模拟量+编码器的 CNC 轴，数字量驱动（EtherCAT，SERCOS，CANOPEN 等）；
- 直线插补、圆弧插补、螺旋插补；
- 龙门轴、主轴功能、刀具功能、侦测功能等；
- 反向间隙补偿、螺距补偿等；
- 宏指令、子程序、语法等；
- DIN66025 编程语言。

CNC 全版本：该版本支持：

- 1 个 CNC 插补通道；
- 8 根 CNC 插补轴，可扩展至 32 轴插补；
- 模拟量+编码器的 CNC 轴，数字量驱动（EtherCAT，SERCOS，CANOPEN 等）；
- 直线插补、圆弧插补、螺旋插补；
- 龙门轴、主轴功能、刀具功能、侦测功能等；
- 反向间隙补偿、螺距补偿等；
- 宏指令、子程序、语法等；
- DIN66025 编程语言。

PACKAGE 软件:

- 1、TwinCAT CNC Axes Pack: 扩展轴, 从 8 轴扩展至 12、32、64 根轴;
- 2、TwinCAT CNC Channel Pack: 扩展通道, 从 1 通道扩展至 12 通道;
- 3、TwinCAT CNC Transformation: 用于坐标转换如 RTCP\TLC\ Kinematics;
- 4、TwinCAT CNC HSC Pack: 用于高速切割功能;
- 5、TwinCAT CNC Spline Interpolation: 用于样条插补 AKIMA spline / B-spline;

四、CNC 性能指标

CNC 最小插补值: 0.0001mm;
 CNC 直线轴长度: -214 米-----+214 米;
 CNC 旋转轴长度: -594 圈---+594 圈;
 CNC 圆半径长度: 0-100000 米;
 CNC 速度范围: 0,000001—3000 米/分;

五、系统硬件组成

标准机床控制系统硬件组成如下:

显示器:

触摸屏

屏幕尺寸: 6.5、10、12、15、19、24 等;

选配: 功能键、按键、急停等可选配



控制器:

嵌入式控制器: CX 系列;

面板控制器: 面板与控制器一体;

紧凑型控制器: 体积小的工控机;

工控机: 标准工控机。



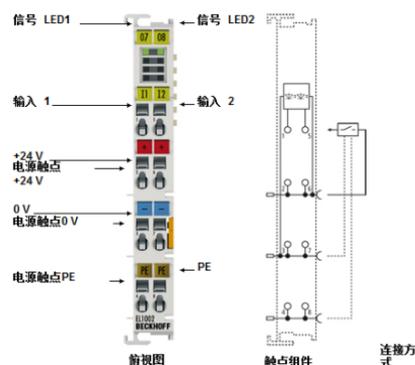
端子:

数字量输入输出端子: EL1004、EL2004

模拟量输入输出端子: EL3102、EL4102

通讯端子: EL6531

特殊端子: EL5101



伺服驱动器：

AX5000 单通道系列：

AX5101\AXS5103\AX5106

AX5112\AX5118\AX5125\AX5140

AX5000 双通道系列：

AX5201\AX5203\AX5206



伺服电机：

AM3000 电机：低惯量电机

AM3500 电机：中惯量电机

AM8000 电机：低惯量电机（编码器动力线编码线单线）

AM8500 电机：中惯量电机（编码器动力线编码线单线）

本章总结：

倍福 TwinCAT CNC 软件是基于 Twincat 的基础上的软 CNC，可以在此基础上开发包括铣床、冲床、磨床等具有各种功能各异的系统，功能完善，满足客户不断变化的需求，灵活且强大。

第二章 TwinCAT CNC 系统软件安装操作

概述:

本章主要介绍 TwinCAT CNC 涉及的软件。倍福 CNC 系统软件分为四个部分,分别为 TwinCAT NCI、TwinCAT CNC、HMIPRO 界面、报警界面软件。对这些软件的安装以及操作进行介绍。
一部分为软件安装

一、TwinCAT NCI 软件安装

安装 CNC 版本的前提是安装了 TwinCAT NCI 软件。至于 TwinCAT 2 的版本没有具体限制,之后, CNC 作为一个插件进行单独安装。与之带来的好处是,我们在使用 CNC 的插补等功能时,可以自由的使用 NC 功能,诸如电子凸轮、电子齿轮、飞剪等功能。

TwinCAT NCI 安装过程在此不赘述!

二、TwinCAT CNC 软件插件

目前倍福的 CNC 软件版本包括 1028E、1028、1030E、1030、1506E、1506、2010E、2010、28 等。
从版本来看,功能升级的几个版本内容如下:

- 1506 版本 (含 1506E): 因应冲床动态特性新增动态数据表格;
- 2010 版本 (含 2010E): 因应激光切割增加尖角控制功能以及小圆功能;

CNC 软件安装:

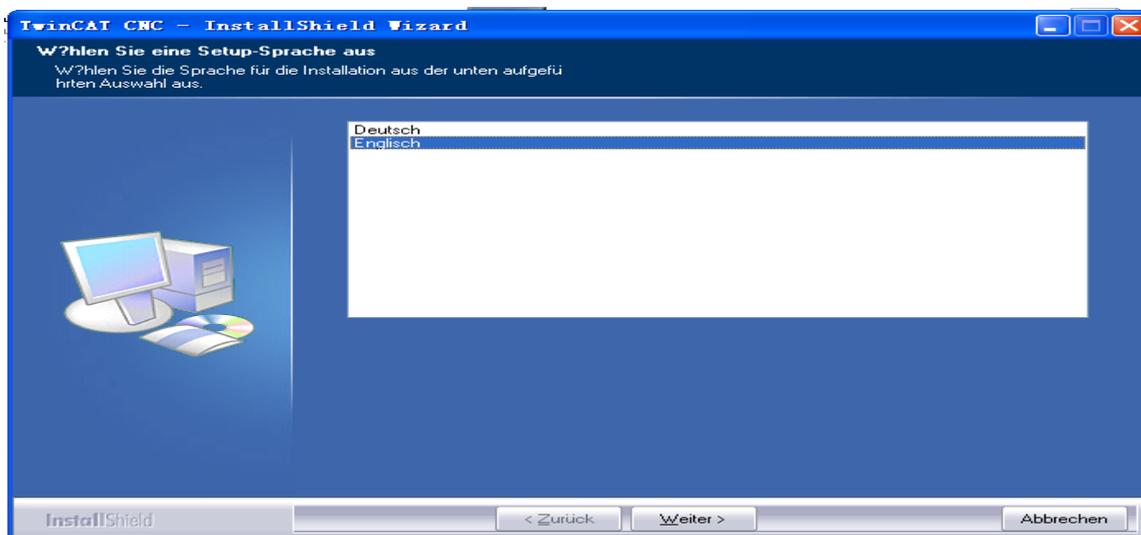
- 1) 安装软件名称及具体信息



TwinCATCnc211_20...
Setup.exe
Macrovision Corp...

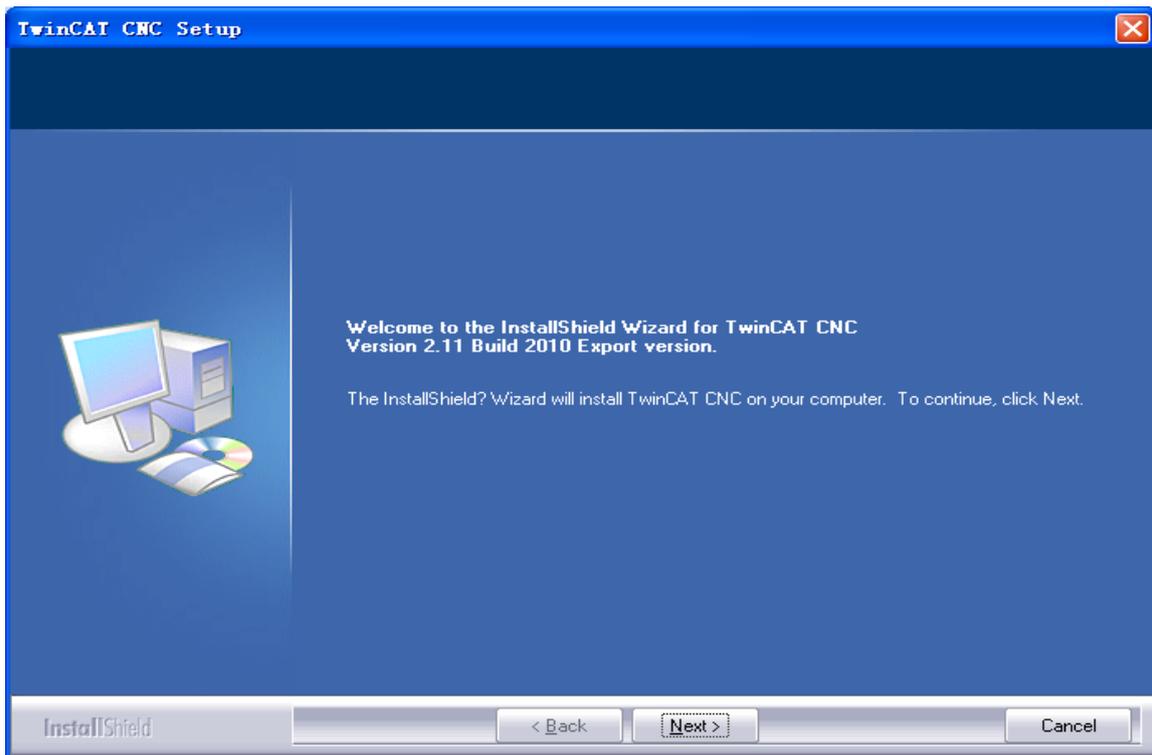
TwinCATCnc211_2010E.exe
描述: Setup.exe
公司: Macrovision Corporation
文件版本: 14.0.0.162
创建日期: 2012-4-9 16:58
大小: 5.75 MB

- 2) 双击安装 TwinCAT 软件, 选择英语安装。



3) 安装 TwinCAT CNC 版本信息

确认当前的版本信息，从示例中我们可以看到当前安装的 CNC 版本为 2010E。确认之后，点 NEXT，执行下步操作。

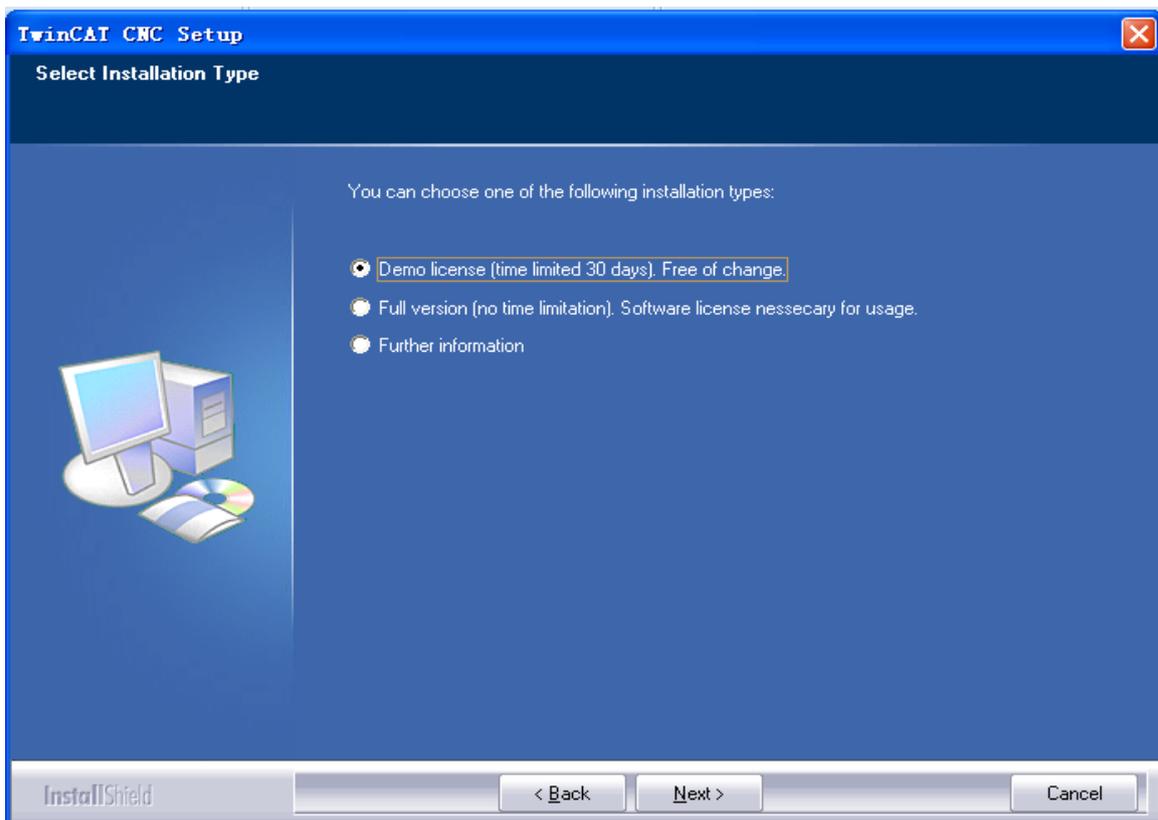


4) 选择安装级别，根据自己的需要选择不同的级别；

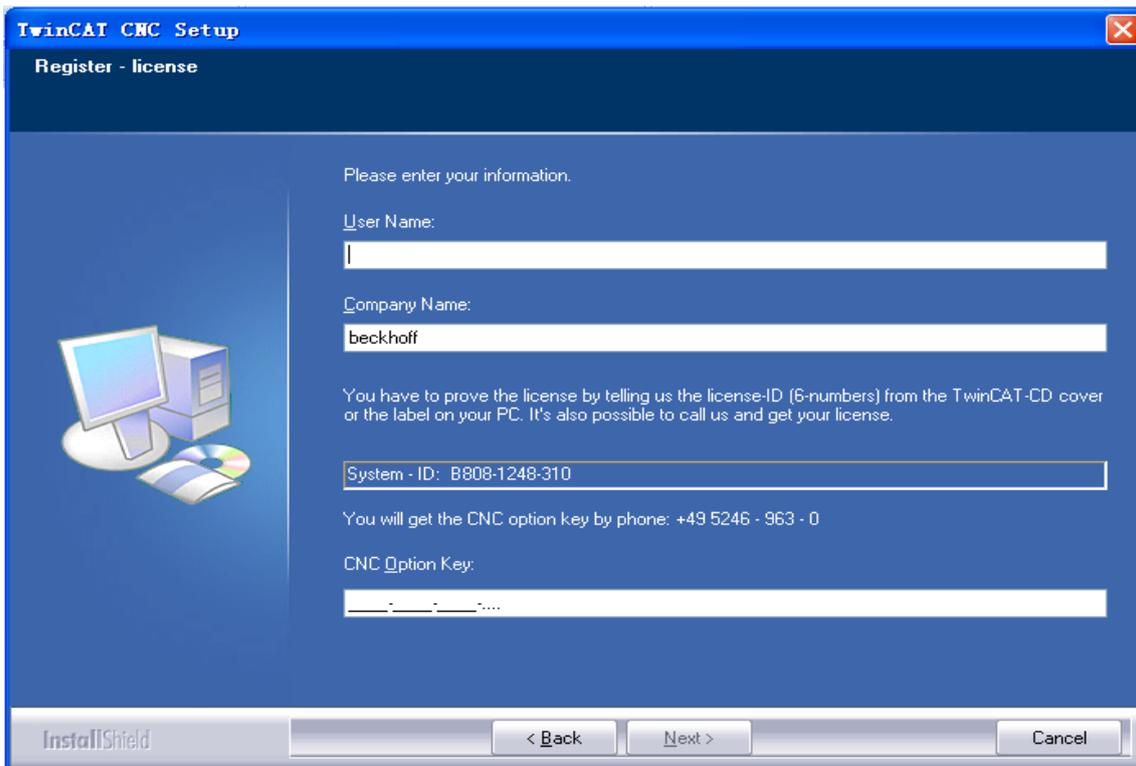
DEMO 级别：该软件使用期限为 30 天，30 天之后需要重新安装才能继续使用；

Full version：安装完成该软件之后，会出现 CNC 版本的唯一 ID，通过该 ID 可以从倍福购买相对应的序列号，已知序列号情况下，可以选择添加序列号 Full version，添加已购买的序列号。

如下图：



下图为填写序列号页面：



5) 结束安装，相应的 CNC 功能就可以使用了，可以发现以下变化：

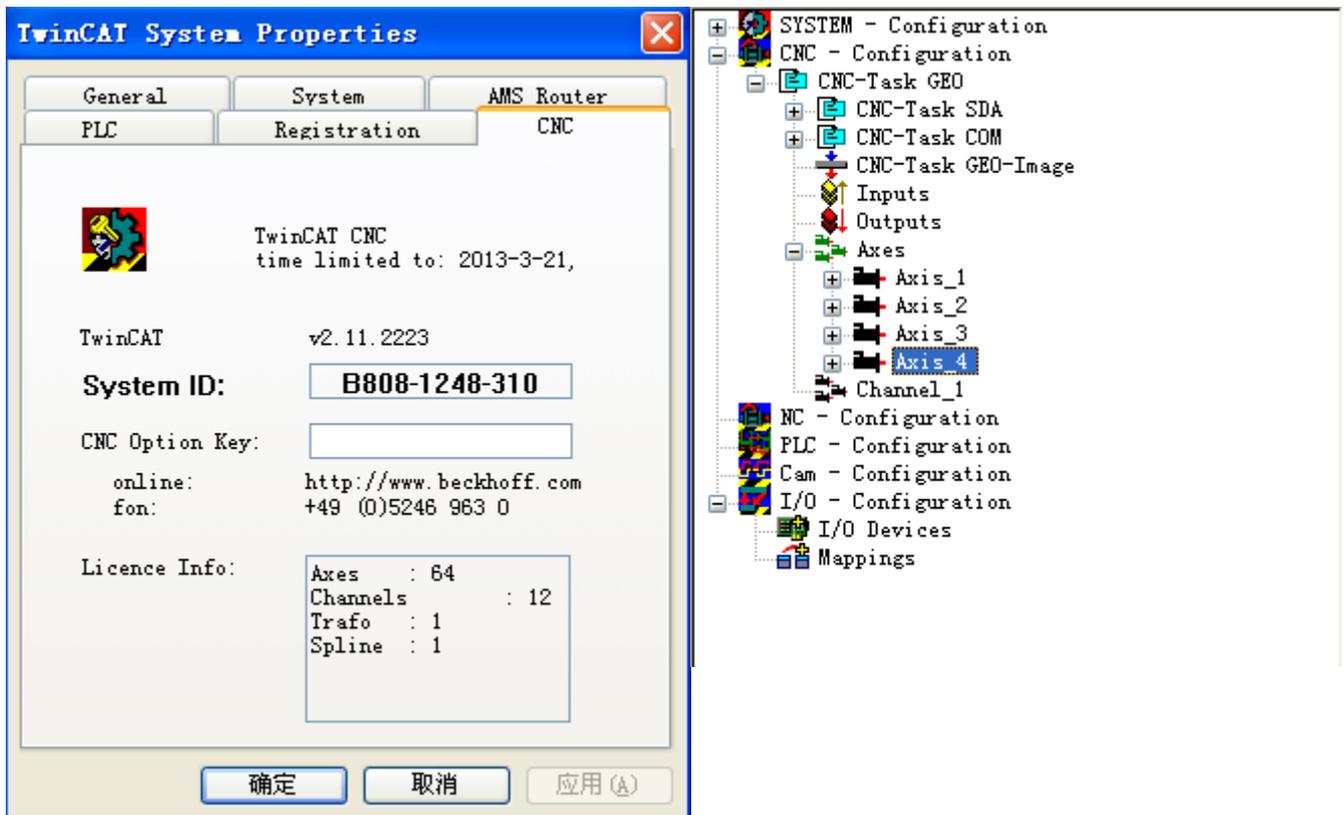
i. TwinCAT 属性文件中，多了个 CNC 选项。

内容：TwinCAT CNC 的使用权限以及版本信息包含系统 ID 及序列号

在图片中你也能够看到你当前的 CNC 级别信息如允许的轴数、通道、主轴、坐标转换等。

ii. System Manager 文件中多了一个 CNC-Configuration 选项，可以在次进行 CNC 功能操作。

重新启动 TwinCAT 软件，CNC 软件生效。



三、CNC 软件更新

倍福的 CNC 系统是一个开放的系统，开放性在于其根据不同的项目应用，经过详细评估沟通可以添加新的 CNC 功能。CNC 的运行核在不断地升级过程中，满足不同项目的差异化需求，体现软 CNC 的特性。

在 CNC 运行核的更新过程中可能需要进行软件的更新。基于此，我们将 CNC 更新方法做了一个简单总结，适用于目前使用的 TWINCAT2.0 下的 CNC 系统，后续的基于 TwinCAT3.0 版本的 CNC 的软件更新将根据情况予以及时更新。

注：具体更新方法可详见[附录一：CNC软件更新](#)！

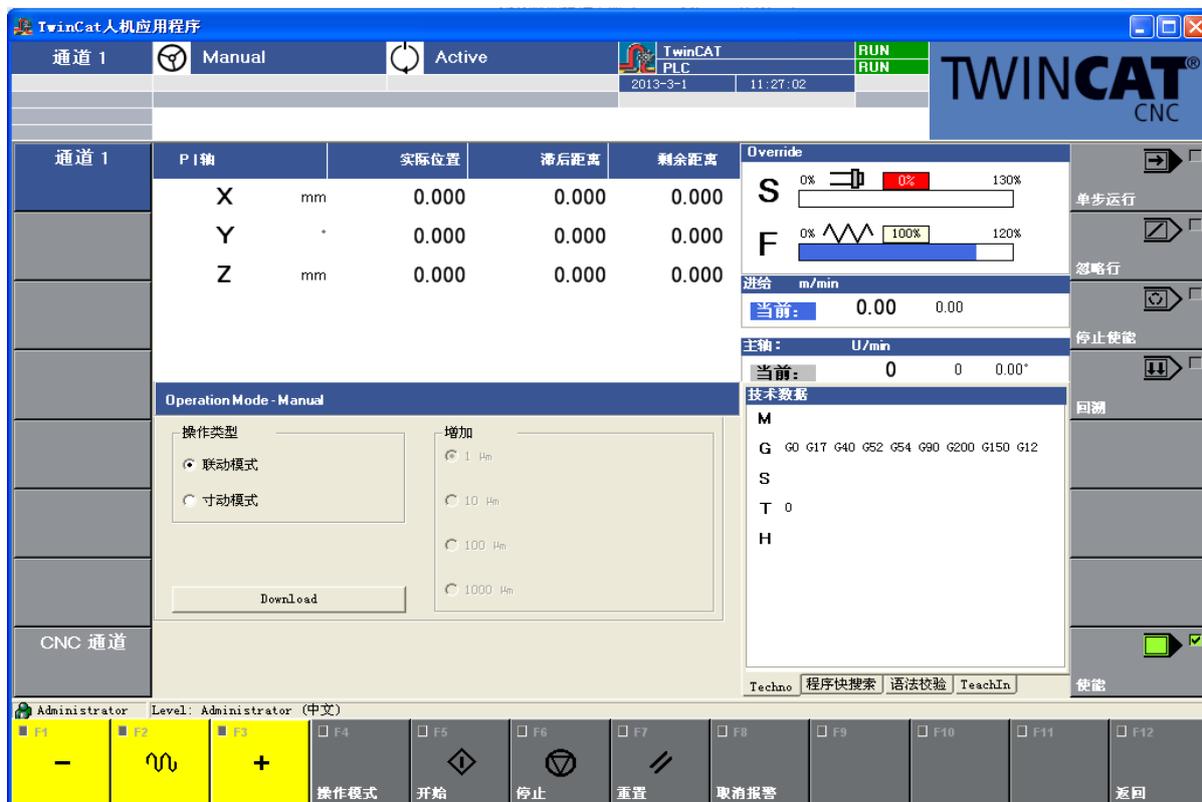
四、CNC 软件界面 HMIPRO 介绍

HMIPRO 界面是倍福基于 C#语言基础上开发的成熟的界面，具有画面美观、操作简单实用、功能全面等优点。倍福对于该软件是全面开放，代码公开，便于客户自主修改。

基本特性如下：

- 1、开放语言开发：基于 VISUAL STUDIO 软件，使用开放的 C#软件开发，具有高校计算机语言的 C#技能的工程师可以方便上手开发；
- 2、基于 ADS 通讯，与 PLC 链接方便，使用变量名等多种方式进行链接；
- 3、界面交互功能，画面不存在复杂的逻辑控制，可另起炉灶重新开发，实现界面差异化；
- 4、目前的画面优点在于功能全面，涵盖了用户级别控制、报警信息、PLC 状态读取、机械参数读写、手动按键添加、控制界面的轴位置显示、G 代码读取等，方便客户直接使用。

注：倍福提供的标准画面介绍可详细参照[附录二：HMI画面介绍](#)。



五、报警信息编辑操作

报警信息自定义，通过 TwinCAT 自带的 eventlog 定义，并显示在界面中，此部分内容将结合 PLC 程序进行讲解。此处列举如何在 eventlog 软件中配置报警信息。

注：倍福提供的标准画面介绍可详细参照[附录三报警信息编辑操作](#)。



六、CAM 软件

CAM 软件在 CNC 的应用过程中，需要将客户工件信息包括形状、大小、位置、图层等信息转换成 CNC 系统识别的 NC 代码。判断一台机床是否能够使用，有一部分工作就是要读取客户的代码，进行判断。

本章总结：

本章主要对于 TWINCAT CNC 软件进行了安装操作方面的讲解，倍福 CNC 软件包含 TwinCAT NCI 软件、CNC 插件、HMIPRO 界面软件以及报警软件组成，所有的操作都是在这个四个软件的基础上进行的。

第三章 TwinCAT CNC 系统编程指令

概述:

BECKHOFF 的 NC 程序结构遵循 DIN66025 标准，是数控系统的规范标准。在机床加工程序中，所有的功能运算都需要由相关指令来触发，加工程序本质上就是由不同的指令字累加起来的集合，从而实现具体的运动控制以及逻辑运算。这里的指令绝大部分是由英文字母、特殊符号以及数字集合出来的。这些指令字也遵循了目前通用指令规范，各家系统不同指令字也相应有些许差异，本章旨在介绍倍福 CNC 系统的系统编程指令，包括 G 代码常用指令以及在 NC 代码中的计算符号与循环指令等。

一、NC 代码介绍

倍福 CNC 代码存储为一个以 .NC 为后缀的文件，保存 G 代码文件时需要保存为这个格式。

单段 NC 程序包括:段号、功能识别字符以及数值或数值表达式等。

✓ G 功能标示符

NC 代码中的 G 功能，用来规定机床与工件的相对运动轨迹、机床坐标系、坐标平面、刀具半径补偿、坐标偏置等，这些功能大致用于路径规划功能以及特殊的测量、补偿功能等，G 指令由字母 G 和后面的数字组成，常见系统包括两个数字，倍福系统提供了一些特殊功能最多包含了 3 个数字，以实现更加灵活的功能控制；

✓ M指令辅助功能

辅助功能主要是控制机床或数控系统的开关功能，如开关冷却液机床外部信号功能，主轴正转、反转、停止、定位以及程序暂停、停止等系统功能。倍福CNC系统最大支持65535个M函数，根据版本不同，TwinCAT CNC目前支持M0-999函数，每个M函数可以根据具体使用赋予不同的属性，以满足实际需求。同时辅助功能H函数最大支持65535个，目前的版本CNC中支持H0-H999。

✓ F、S、T功能指令

1) F指令

F指令是指系统给定进给速度指令，通常直接给定速度值，如F1000，即速度为1000mm/min；

2) S指令

S指令是指主轴转速指令，直接给定速度值，不同于F指令的是，S指令速度值为转/分，如S1000，即为主轴转速为1000转/分，搭配M函数执行M03\M04\M05M19等；

3) T、D指令

T指令是用于指定加工刀具号，用于具有刀具的加工机床，通常搭配D指令执行。实现相应刀具的切换。

✓ 其他指令

1) 循环功能块:

循环指令是指在 G 代码执行过程中根据实际要求进行循环执行的指令，循环指令减少了大量的重复执行代码，实现正常的指令无法做到的无限条件循环等功能。倍福的循环指令非常丰富，如下：

条件跳转: \$IF, \$ELSEIF, \$ELSE, \$ENDIF,

\$SWITCH, \$CASE, \$DEFAULT, \$ENDSWITCH, \$BREAK

累加循环: \$FOR, \$ENDFOR, \$CONTINUE, \$BREAK

执行条件循环: \$WHILE, \$ENDWHILE, \$CONTINUE, \$BREAK

无执行条件循环: \$DO, \$ENDDO, \$CONTINUE, \$BREAK

同NC文件跳转: \$GOTO

2) 数学表达式

数学表达式在G代码中，可以用于条件判断也可以用于轴位置的内部计算，可用的数学公式如下：

标准算术操作：+ , - , * , / , ** , MOD

数学表达式：ABS,SQR,SQRT,EXP,LN,DEXP,LOG

角度表达式：SIN,COS,TAN,ASIN,ACOS,ATAN

单位换算：INT,FRACT,ROUND

数学表达式：

数学符号、数字、功能参数、变量等都可以在表达式中体现，示例如下：

```
[[sin["MAX_ANZ" * 30.00] + P2] / V.G.SATZ_NR]
```

3) 变量定义，

NC 代码中可以定义本地变量、全局变量等变量类型，并包括系统内部变量与外部变量。

系统内部变量主要针对轴变量 V.A.NAME 以及通道变量 V.G.NAME，通过 V.A.NAME，V.G.NAME 访问 CNC 内部的数据诸如，各轴位置、测量数据、刀具数据、偏置等；

本地变量与全局变量 V.L.name、V.S.name、V.P.name 用于系统变量声明便于 G 代码逻辑运算；

外部变量 V.E.NAME 可以直接通过 PLC 程序实时读取和写入，执行相应的系统功能；

4) CNC 内部变量，特殊功能符

条件忽略行 ‘/’，在 CNC 控制中执行“跳行”功能，可以选择执行使用该功能的注释行。

5) 段号 N 值：

该值用于指示当前行号，系统可以从 HLI 中读出当前系统行号并进行操作。在 NC 代码文件中，可以按照需要不使用、部分使用、每行代码都是用的方式。

without numbering	partial numbering	complete numbering
% 100	% 100	% 100
"Block 1"	N10 "Block 1"	N10 "Block 1"
"Block 2"	"Block 2"	N20 "Block 2"
"Block 3"	N20 "Block 3"	N30 "Block 3"
.	"Block 4"	N40 "Block 4"
.	.	.
.	.	.
M30	M30	N700 M30

6) 注释内容

注释符号：“(”

注释内容几乎可以插入一个 NC 程序的任一位置，包括程序头。注释内容不影响 NC 运行。如果在程序段结尾单独注释可以只使用“(”，在程序行中间注释需要在结尾加注“)”。另一种方法使用“;”，这种方式总是放在本行结束位置。允许嵌套注释。

% 100 (Comment in complete brackets)

N200 ... (Comment only with open bracket

N300 (Comment (Nested comment))

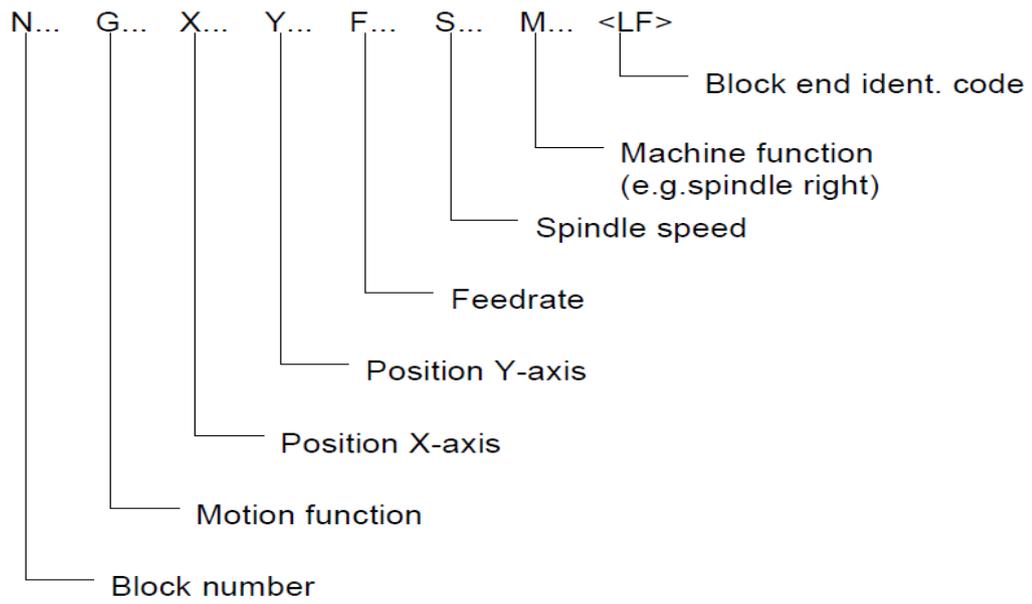
N500 X10 (Comment within a block) Y20

N700 ... ;Comment behind semicolon

N999 M30

✓ NC 代码指令示意

一个NC程序段包括三部分程序段号、程序段、程序结束标志。



二、G 代码介绍

G 功能中描述的包括移动方式（绝对、相对）、插补方式（直线、圆弧、螺旋）、测量功能、时间相关以及激活操作状态等。G 代码的类型包括模态、非模态等类型，不同的 G 代码组合使用将在下列程序中使用。本文将根据 G 代码的具体功能分类进行讲解，包括插补指令、坐标系指令、参考点指令、刀具补偿指令、子程序指令、辅助指令等。

1、范例程序

```
% 100 (program example)

N10 G00 X100 Y100 (Rapid traverse mode of the X- and Y-axes)(to position 100)

N20 Z100 (Rapid traverse mode of the Z-axis to 100.)

      (G00 remains active till deselection by)(another G-function.)

N30 G1 Z50 F5 S1000 M3 (Spindle right-hand side rotation (1000 rpm)

      (and linear interpolation with feedrate)((5 m/min) to position Z=50)

N40 Z100 (Linear interpolation to Z=100)

N50 G0 X200 Y200 Z200 (Rapid traverse to position X=Y=Z=200)

N50 M5 (Spindle stop)

N60 M30 (Program end)
```

基本功能：

G00 快速移动；

G01 直线插补；

主轴功能：

S1000 主轴转速 1000rpm；

进给速度：

F5 进给速度 5mm/min；

控制轴：

X, Y, Z 进给轴 XYZ；

机床功能：

M03 主轴正转；

M05 主轴停止；

M30 程序结尾；

2、G 代码模式

程序地址 G 表示准备功能，通常称为 G 代码，它用来指令机床进行加工和插补功能。按照运行性质可以分为两种：

模态 G 代码：一旦使用了这样的指令，则一直有效，知道被同组的其他 G 代码取代为止；例如：G90 为绝对位置，G91 为相对位置。当执行了 G90，G 代码会一直为绝对位置移动，除非遇到 G91，才能够执行相对移动。

非模态 G 代码：只在被调用的程序段有效。例如 G04 延时功能，只在调用的程序有效。

3、G 代码功能符

G功能符	功能表述
G00	Rapid traverse(P44)
G01	Linear interpolation(P45)
G02	Circular interpolation,clockwise(cw.) (P46)
G03	Circular interpolation, counter-clockwise (ccw.) (P46)
G02/03	Helical interpolation(P50)
G04	Dwell(P57)
G05	Direct tangential selection/deselection of TRC (P131,370)
G08	Acceleration at block begin(P78)
G09	Deceleration at block end(P78)
G10	<i>Feedrate constant (P132,370)</i>
G11	<i>Feedrate adapted (P132,370)</i>
G12	Deselection of corner feed reduction(P130)
G13	Selection of corner feed reduction(P130)
G17	X-Y-plane(P83)
G18	Z-X-plane(P83)
G19	Y-Z-plane(P83)
G20	Deselection of mirroring(P84)
G21	Mirroring of programmed path on the Y-axis(P84)
G22	Mirroring of programmed path on the X-axis(P84)
G23	Superimposing of G21 and G22(P84)
G25	Linear transitions for TRC(P131,370)
G26	Circular transitions for TRC(P131,370)
G33	Thread cutting, uniform lead(P450)
G40	TRC/SRK deselection (P131,370,443)
G41	TRC/SRK left of the contour(P131,370,443)
G42	TRC/SRK right of the contour(P131,370,443)
G51	Selection of diameter programming(P445)
G52	Deselection of diameter programming(P445)
G53	Deselection of zero offsets(P133)
G54...G59	Selection of zero offsets(P133)
G60	Exact stop(P95)
G61	Polynom contouring(P96)
G63	Tapping(P452)
G70	Inch data input(P91)
G71	Metric data input(91)
G74	Homing(P58,464,473)
G80–G89	Implicite subprogram calls(P92)
G90	Absolute dimensioning(P92)
G91	Incremental dimension (relative)(P92)
G92	Coordinate preset (P59)
G93	F-word for specifying machining time(P155)
G94	F-word for specifying the federate(P155)
G95	Feedrate in millimeter per revolution(P447)
G96	Constant cutting speed(P448)
G97	Specification of spindle speed in 1/min(P448)

G98	Set negative software limit switch(P60)
G99	Set positive software limit switch(P61)
G100	Measuring functions (Typ 1- 7) (P 62)
G101	Inclusion of measuring offsets in shifts (P 72)
G102	Extracting measuring offsets from shifts (P 72)
G106	Measurement with movement to target point(P 66)
G107	Deselection of block global edge banding (P 76)
G108	Edge banding (P 74)
G112	Gear change (P 173, 463)
G115	General influencing of look-ahead (P 174)
G116	Manipulate look-ahead (P 174)
G117	Switch on look-ahead (default (P 174)
G129	Weighting of rapid traverse velocity (P 151)
G130	Axis-specific weighting of acceleration (P 152)
G131	Axis group-specific weighting of acceleration (P 152)
G132	Axis-specific weighting of ramp time (P 153)
G133	Axis group-specific weighting of ramp time (P 153)
G134	Axis group-specific weighting of geometrical ramp time (P 153)
G135	Selection of feedforward control (P 150, 476)
G136	Specification of weighting the feedforward control (P 150, 476)
G137	Deselection of feedforward control (P 150, 476)
G138	Direct selection/deselection of TRC (P 131, 370)
G139	Indirect selection/deselection of TRC (P 131, 370)
G140	Deselection of contour masking (P 132, 370)
G141	Selection of contour masking (P 132, 370)
G150	Deselection of spline interpolation (P 282, 287)
G151	Selection of spline interpolation (P 282, 287)
G159	Extended zero offsets (P 138)
G160	Axis specific activation of zero offsets (P 139)
G161*	Circle center point, absolute (P 140)
G162**	Circle center point, relative (P 140)
G163	Radius programming (P 141)
G164	Circle center point offset off (P 144)
G165	Circle center point offset on (P 144)
G166	Override 100% (P 177)
G167	Spindle override 100% (P 465, 473)
G193	Path related feed adaptation (P 81)
G194	Weighting of maximum feed velocity (P 155)
G196	Maximum spindle speed for G96 (P 448)
G200	Selection of manual mode without parallel interpolation (P 164)
G201	Selection of manual mode with parallel interpolation (P 163)
G202	Deselection of manual mode with parallel interpolation (P 163)
G237	Perpendicular selection/deselection of TRC (P 131, 381)
G238	Inner corner selection of TRC (P 131, 388)
G239	<i>Direct selection/deselection of TRC without a block/</i> (P 131, 391)
G260	Deselection of polynom contouring (P 96)
G261	Selection of polynom contouring (P 96)
G293	Time related feed adaptation (P 81)
G301	Inserting of chamfers (P 156)

G302	Inserting of rounding (P 156)
G310	Interruptible block (Measurement typ 5 - 6) (P 70)
G351	Mirroring with axis information (P 89)
G359	Deselection of exact stop (P 95)
G360	Selection of exact stop (P 95)
G800	– G819 Additional implicate subprogram calls (P 92)
G900	Switching to deceleration after block end (P 78)
G901	Deceleration after block end (P 78)

*If G162 is active:

- I:Relative position of the circle center point in X-direction
- J:Relative position of the circle center point in Y-direction
- K Relative position of the circle center point in Z-direction

**If G161 is active:

- I Absolute position of the circle center point in X-direction
- J Absolute position of the circle center point in Y-direction
- K Absolute position of the circle center point in Z-direction

三、G 代码插补指令

G 代码插补指令大致包括快速定位指令 G00、直线插补指令 G01、顺时针圆弧（包括顺时针螺旋线）插补指令 G02、逆时针圆弧（螺旋线）插补指令 G03 组成。

1、快速定位指令 G00

1.1 指令格式：

G00 X[POS] Y[POS]

1.2 指令说明：

G00 为快速移动，模态指令；

1.3 指令模式：

可以使用绝对运动 G90、相对运动 G91 ；

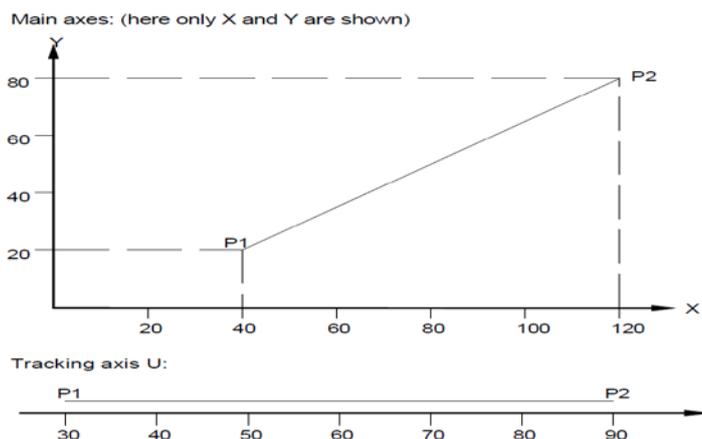
1.4 指令参数：

G00 速度加速度值是由各轴配置文件中的 G00 参数决定的，与机床的进给速度无关，可以通过倍率开关来实现速度调节；

1.5 指令轨迹：

利用 G00 使刀具快速移动，在倍福系统中，各轴同时从起点到达终点，各轴中至少有一个轴到达 G00 设定速度。

示例如下：



示例NC代码：

绝对位移：

Nnnnn G00 G90 X120 Y80 U90 (traverse from P1 to P2)

相对位移：

Nnnnn G00 G91 X80 Y60 U60 (traverse from P1 to P2)

1.6 圆弧指令：

当 G00 G90 控制旋转轴时，旋转轴会按照模长执行，即执行的最大长度不超过半圈。

1.7 指令相关

G00 运行的轨迹为折线，为了使刀具在一栋过程中，避免发生撞车，在编写 G00 时，建议将 X、Y 与 Z 轴分开写。

当刀具需要接近工件时，先 X、Y 后写 Z，即：

G00 X_ Y_
Z_

当刀具需要远离工件时，先 Z 后 X、Y，即：

G00 Z_
X_ Y_

2、直线插补指令 G01

2.1 指令格式:

G01 X[POS] Y[POS] F[SPEED]

2.2 指令说明:

G00 为直线插补指令，是模态指令；

2.3 指令模式:

可以使用绝对运动 G90、相对运动 G91。当采用绝对模式时，坐标值是相对于工件坐标系的原点；采用增量模式是相对于刀具当前位置的坐标。

2.4 F_进给速度:

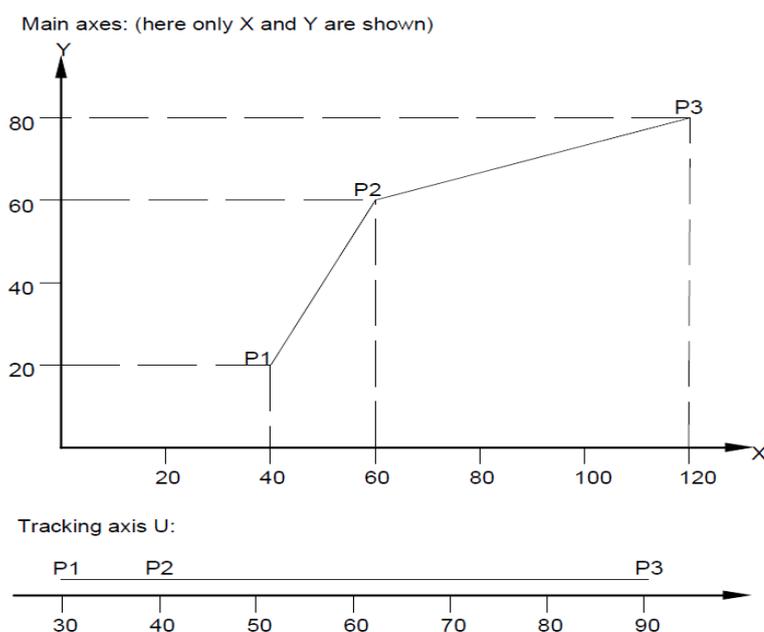
进给速度是模态指令。在开始直线插补之前，程序中必须包含有效的进给率，否则在启动运行时会出现报警，也可以在配置文件中设置不给定插补速度的默认速度值。G01 直线插补的加速度值是由各轴配置文件中的 G01 参数决定的，可以通过倍率开关来实现速度调节；

速度的单位通常设定为 $\text{mm}(\text{°})/\text{min}$ 。

1.5 指令轨迹:

直线插补中各轴，必须保证各轴同时按照直线插补，直线轴中各轴的分速度与各轴的移动距离成正比，以保证指令各轴同时到达终点。

示例如下:



示例NC代码:

绝对位移:

```
Nnn G01 G90 X60 Y60 U40 F1000 (traverse from P1 to P2 feedrate)
(1000mm/min)
```

```
Nnn X120 Y80 U90 (traverse from P2 to P3 feedrate 1000mm/min)
```

相对位移:

```
Nnn G01 G91 X20 Y40 U10 F1000 (traverse from P1 to P2 feedrate)
(1000mm/min)
```

```
Nnn X60 Y20 U50 (traverse from P2 to P3 feedrate 1000mm/min)
```

2、圆弧插补指令 G02\G03

圆弧插补指令用于编写圆弧或完整的圆，其中顺时针圆弧插补指令 G02，逆时针圆弧（螺旋线）插补指令 G03。

2.1 指令格式：

G02 G03 [X<expr> Y<expr>] I<expr> J<expr> R<expr>
--

G02 G03	Circular interpolation CW / CCW
X<expr> Y<expr>	End point in XY plane
I<expr> J<expr>	Position of circle center point of the interpolation in XY plane (I in X, J in Y), according to G161/G162
R<expr>	Radius of the circle (alternative to I,J)

在 XY 平面上：

$$G17 \begin{matrix} (G02) \\ (G03) \end{matrix} X * Y * \begin{pmatrix} R^* \\ I^*J^* \end{pmatrix} F *$$

在 XZ 平面上：

$$G18 \begin{matrix} (G02) \\ (G03) \end{matrix} X * Z * \begin{pmatrix} R^* \\ I^*J^* \end{pmatrix} F *$$

在 YZ 平面上：

$$G19 \begin{matrix} (G02) \\ (G03) \end{matrix} Y * Z * \begin{pmatrix} R^* \\ I^*J^* \end{pmatrix} F *$$

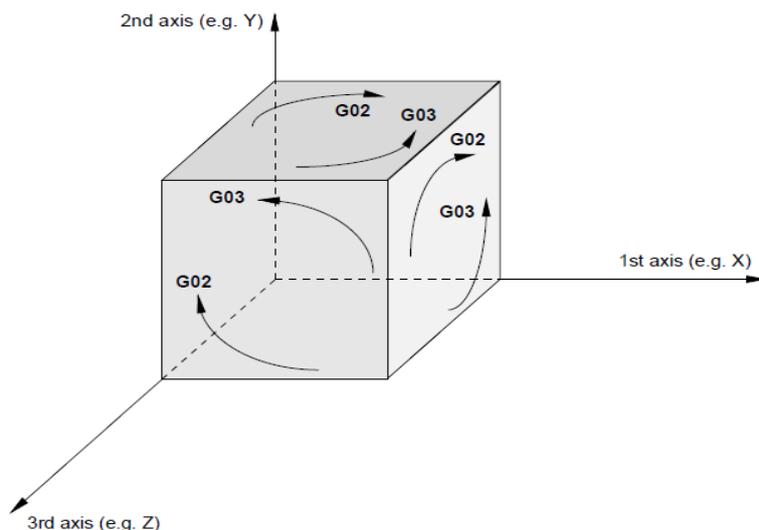
2.2 指令说明：

G02 是顺圆插补指令，G03 逆圆插补指令，均为模态指令；

2.3 指令平面选择：

圆弧加工必须有平面选择，G17、G18、G19 为圆弧插补平面选择指令，以此来确定被加工表面所在平面，如果在 XY 平面，G17 指令可以省略。

下图是圆弧在各平面的插补方向指示。



2.4 指令参数：

圆弧终点坐标：

X_Y_Z 表示圆弧终点坐标，可以使用绝对坐标系和相对坐标系编程，采用绝对坐标系时，终点坐标是相对于原点的；采用增量模式时，终点坐标是相对于圆弧起点来讲的；当 XYZ 值不变时，可以省略，此时起点和终点位置重合，此时的圆弧为整圆。

2.5 圆心坐标:

I、J、K_表示圆弧圆心的坐标偏置量，无论使用 G90 还是 G91，它是圆心位置不影响，当设定 G161 时，圆心位置采用绝对值方式，G162 时，采用相对值方式。G17 平面为 I, J, G18 平面为 I、K, G19 平面为 J, K。当 I、J、K 为 0 的时候可以省略。

圆心坐标偏置方法: KM: 圆心

G162:	(basic settings)
I	- relative position of Km in the X-direction
J	- relative position of Km in the Y-direction
K	- relative position of Km in the Z-direction
G161:	
I	- absolute position of Km in the X-direction
J	- absolute position of Km in the Y-direction
K	- absolute position of Km in the Z-direction

2.6 圆弧半径 R

圆弧半径 R，有正负之分，过起点和终点的圆弧有两个，即小于 180° 的圆弧和大于 180° 的圆弧。为了区分哪一个圆弧，对于小于 180° 的圆弧，半径采用正值表示；对于大于 180° 的圆弧，半径采用负值表示；对于等于 180° 的圆弧，半径用正负值均可。

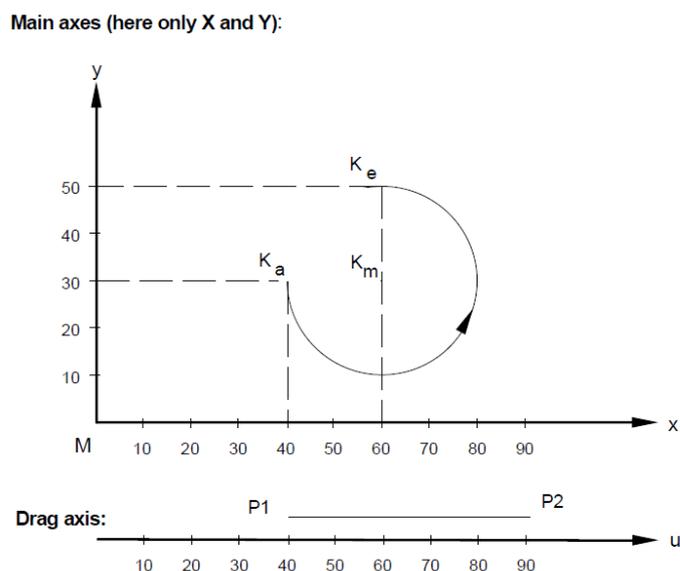
2.7 指令速度

指令圆弧进给速度为圆弧的切向进给速度。单位为 mm/min。

2.8 圆度限制

最大的圆半径不能超过 10^9mm ，圆弧终点位置不能超过从起点开始的 $\pm 2.14 \times 10^5\text{mm}$ 。

代码图片:



示例代码:

绝对位移:

Nnn G90 F1000 (Absolute dimension, feedrate)

Nnn G17 (Selection of X-Y-plane)

Nnn G03 G161 X60 Y50 I60 J30 U90 (Circle: Ka -> Ke and

(linear interpolation: P1 -> P2)

相对位移:

Nnn G91 F1000 (Incr. dimension, feedrate)

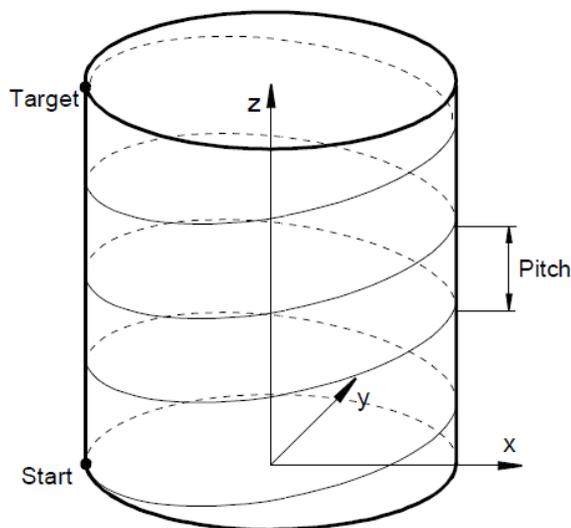
Nnn G17 (Selection X-Y-plane)

Nnn G03 G162 X20 Y20 I20 U50 (Circle : Ka -> Ke and)

(linear interpolation: P1 -> P2)

3、螺旋线插补

螺旋线插补是在圆弧插补的基础上，进行的第三根轴方向上的直线位移。螺旋上升第三轴方向必须要维持一个固定值。第三方向取决于圆弧所在的坐标平面。



3.1 指令格式:

Syntax example for plane G17:

G02 G03 X<expr> Y<expr> Z<expr> I<expr> J<expr> R<expr> K<expr>
--

G02 G03	Circular interpolation CW / CCW
X<expr> Y<expr>	Target point in plane XY
Z<expr>	Target point on the helical axis perpendicular to plane XY
I<expr> J<expr>	Position of the circle center point of the interpolation in plane XY (I in X, J in Y), corresponding to G161/G162.
R<expr>	Radius of the circle to be interpolated (alternative to I,J)
K<expr>	Helix pitch in Z (value generally <u>unsigned</u>)

在 XY 平面上:

$$G17 \begin{matrix} (G02) \\ (G03) \end{matrix} X * Y * \begin{pmatrix} R^* \\ I^* J^* \end{pmatrix} Z * F *$$

在 XZ 平面上:

$$G18 \begin{matrix} (G02) \\ (G03) \end{matrix} X * Z * \begin{pmatrix} R^* \\ I^* J^* \end{pmatrix} Y * F *$$

在 YZ 平面上:

$$G19 \begin{matrix} (G02) \\ (G03) \end{matrix} Y * Z * \begin{pmatrix} R^* \\ I^* J^* \end{pmatrix} X * F *$$

3.2 指令说明:

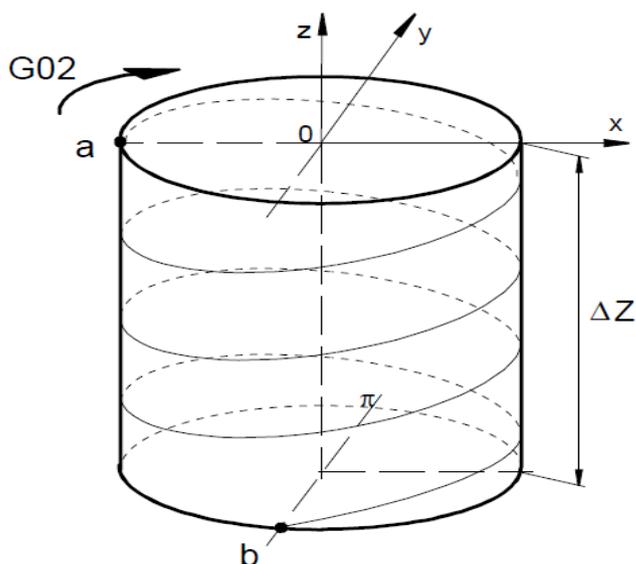
螺旋线插补是工作平面内两周联动的圆弧插补运动与另一轴上的直线运动。三根轴的运动通常是同步的，并且同时到达目标位置。

3.3 指令模式:

螺旋插补的插补运动格式遵循螺旋插补运动原则,选择的平面模式,决定了哪根轴有效以及它们的功能,下表列出了平面选择与各轴之间的关系。

Plane	Type of interpolation	Target point in plane	Target point on helical axis	Center /radius	Pitch
G17	G02/G03	X..Y..	Z..	I..J../R	K
G18	G02/G03	Z..X..	Y..	K..I../R	J
G19	G02/G03	Y..Z..	X..	J..K../R	I

示例如下:



示例代码:

Helical interpolation clockwise in plane XY with following helix:

Starting point a: X-10 Y0 Z0

Target point b: X0 Y-10 Z-20

Helix center I, J: zero point

Helix pitch K: variable:

N10 G17 G90 X-10 Y0 Z0 F500

N20 G02 X0 Y-10 Z-20 I0 J0 K..

4、各轴寻参考点 G74

4.1 指令格式:

G74 X2 Y2 Z1

4.2 指令说明:

G74 指令控制各轴按照相应顺序执行回参动作,回参顺序取决于各轴名称后对应的数字顺序,对于各轴后数字大小相同的回参,同时执行回参动作。

4.3 指令模式:

G74 回参操作包括回参速度设定,编码器类型,动作方式等设定。设定的参数以及执行状态具体见第五章,回参功能。

四、坐标值与尺寸

1、英制和公制单位选择 G70/G71

1.1 指令格式:

G70

G71

1.2 指令说明:

G70/G71 为机床运行尺寸，是模态指令；

G70: 选择英尺单位

G71: 选择米制单位

默认使用 G71，米制单位。

1.3 指令模式

建议在同一系统中，只使用一种单位系统，并在系统开始的时候即选择相应的单位系统。

2、绝对模式和增量模式 G90-G91

2.1 指令格式:

G90 X_ Y_

G91 X_ Y_

2.2 指令说明:

G90 为绝对坐标编程，坐标值是相对于原点来定义的；

G91 为增量坐标编程，又称为相对坐标编程，坐标值是相对于当前位置来定义的；

G90\G91 是模态指令，彼此可以相互取消；控制系统使用的初始缺省设置是 G90

2.3 指令模式

同一程序段中，可以混合使用绝对和增量模式，实现相应的运动控制。

2.4 重复使用 G90/G91

在默认模式下，同一行代码中不能出现重复使用 G90/G91 代码。



Programming example

```

;
N10 X10 Y10 (Absolute dimensioning G90 is selected, Default)
N20 G91 X20 Y20 (Deselection of absolute- and selection of relative)
                    (programming)
N30 X30 G90 Y30 (Deselection of relative- and selection of absolute)
                    (programmig)
;
N100 G90 Z30 G90 (Error message: Redundant programming of G90/G91)
N110 G91 X10 G90 Z30 (Error message: Redundant programming of G90/G91)

```

设定 P-CHAN-00116 可实现同一行代码中，同时使用 G90\G91，

参数如下：

Variable name	Type	Permitted range	Dimension
multi_dimension_in_block	BOOLEAN	0/1	----

0	G90/G91 may be programmed only exclusively in the same NC block. (Default).
1	G90/G91 may be programmed at the same time in the same NC block. The dimension programmed last is valid for all following positions up to the next G90/G91.

代码如下：



Programming example

```

;
N10 X10 Y10 (Absolute dimensioning G90 is selected, Default)
N20 G91 X20 G90 Y20 (Relative for X axis, absolute for Y axis)
N30 X30 G91 Y30 Z20 (Absolute for X axis, relativ for Y/Z axes)
N30 G90 X30 G91 Y30 G90 Z20 (Absolute for X/Z axes, relative for Y axis)
;
N100 G90 G91 Z30 (Relative for Z axis)
N110 G91 X10 G90 Z30 (Relative for X axis, absolute for Z axis)

```

五、坐标系偏置指令

1、平面选择 G17、G18、G19

1.1 指令格式:

G17

G18

G19

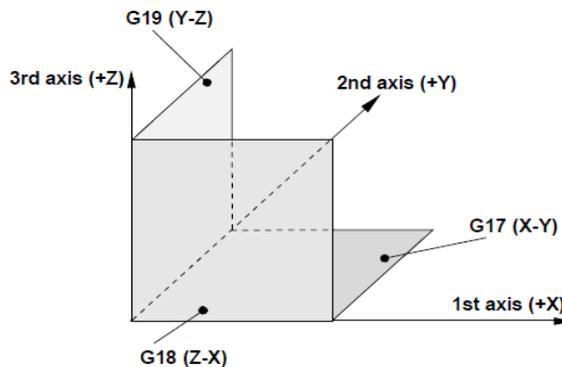
1.2 指令说明:

G17、G18、G19 为平面选择指令，是模态指令，默认为 G17；

G17 为 XY 平面，X 为第一轴，Y 为第二轴；

G18 为 ZX 平面，Z 为第一轴，X 为第二轴；

G19 为 YZ 平面，Y 为第一轴，Z 为第二轴；



1.3 指令模式:

平面选择指令影响圆弧插补、刀具半径补偿，所以在程序中必须含有平面选择指令，一般讲平面选择指令卸载加工运动指令的前面，通常设定为 G17，可以省略不写；

半径补偿：半径补偿通常作用在坐标平面的前两根轴；

圆弧插补：选择坐标平面后，X、Y、Z 对应的半径 I、J、K 发生变化，下图显示半径选择：

Plane	Interpolation type	End point in plane	Center point /Radius
G17	G02/G03	X..Y..	I..J../R
G18	G02/G03	Z..X..	K..I../R
G19	G02/G03	Y..Z..	J..K../R

2、坐标系偏置指令 G92

1.1 指令格式:

G92 X_ Y_

1.2 指令说明:

G92 是建立工件坐标系指令，属于非模态指令；



Programming example

```

N10 G90 (Absolute dimensional specification)
N20 G92 X10 Z30 (Displaces the programmed and the absolute)
. (coordinates by 10 in X, 30 in Z)
.
.
Nnn G92 X0 Z0 (Reset of coordinate preset to 0)

```

3、坐标系偏置 G53-G59/G159

3.1 指令格式

G54 (~G59)

3.2 指令说明

G54~G59 为工件坐标系，G54 为数控机床的第一坐标系，以此类推，为第二至第六坐标系；通常在程序的开头需要指定工件坐标系；

G54~G59 为模态指令，G53 取消坐标系偏置。

3.3 指令模式

G53-G59 坐标系设定在 CNC 中的 NPV 参数设置。

G159 可以指定相应的坐标系偏置，如下：

```
G159 = 0      corresponds to G53
G159 = 1      corresponds to G54
G159 = 2      corresponds to G55
G159 = 3      corresponds to G56
G159 = 4      corresponds to G57
G159 = 5      corresponds to G58
G159 = 6      corresponds to G59
G159 = 7      access to data record with list index 7
:
G159 = 10     access to data record with list index 10
:
```

3.4 指令设定

Variable	Significance	Read	Write
V.G.NP[j].ALL	Addressing of all axes of a zero offset	yes	yes
V.G.NP_AKT.V[j]	Current (currently active) zero offset of an axis	yes	yes
V.G.NP_AKT.ALL	Addressing of the current (currently active) zero offsets of all axes	yes	yes
V.G.NP_DEFAULT	Zero offset that is effective in the initial state.	yes	yes

NPV1 对应 G54，NPV2 对应 G55

N10 V.G.NP[1].ALL = V.G.NP[2].ALL

N20 V.G.NP[1].V.X = V.G.NP[4].V.X + 100



Programming example

The current zero offset of the X axis shall be 200.

```
N10 V.G.NP_AKT.V.X = 200
```

The current zero offset in all axes shall be expanded by the offset values from G55 (NPV2).

```
N10 V.G.NP_AKT.ALL = V.G.NP_AKT.ALL + V.G.NP[2].ALL
```

or

```
N10 V.G.NP_AKT.ALL += V.G.NP[2].ALL
```

六、G 代码特殊功能指令

1、延时功能 G04 | #TIME

1.1 指令格式:

G04 X_

1.2 指令说明

G04 延时功能为非模态指令，延时时间设定可以使用第一轴的功能符加时间设定或者直接设定时间，单位为秒（S）。



Programming example

```
N10 G04 X4.13          (wait for 4.13 seconds)
or
N20 G04 2.13          (wait for 2.13 seconds)
```

1.3 宏指令#TIME

与 G04 的用法相同。



Programming example

```
N10 #TIME 4.13        (wait for 4.13 seconds)
```

2、刀具补偿功能

● 刀具补偿指令 G40/G41/G42

2.1 指令格式

在 XY 平面上:

$$G17 \begin{cases} G40 \\ G41 \begin{pmatrix} G00 \\ G01 \end{pmatrix} X * Y * F * \\ G42 \end{cases}$$

在 XZ 平面上:

$$G18 \begin{cases} G40 \\ G41 \begin{pmatrix} G00 \\ G01 \end{pmatrix} X * Z * F * \\ G42 \end{cases}$$

在 YZ 平面上:

$$G19 \begin{cases} G40 \\ G41 \begin{pmatrix} G00 \\ G01 \end{pmatrix} Y * Z * F * \\ G42 \end{cases}$$

2.2 指令说明

G41 为刀具半径左补偿，属于模态指令，刀具沿前进方向，向左侧进行补偿；

G42 为刀具半径右补偿，属于模态指令，刀具沿前进方向，向右侧进行补偿；

G40 为取消半径补偿，属于模态指令；

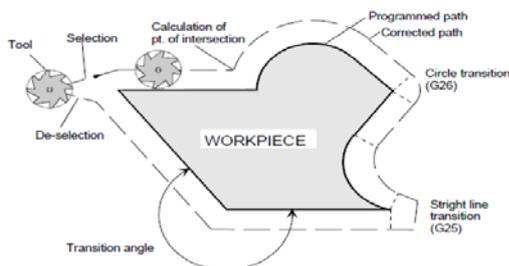


Fig. 12.2: Mechanism and terminology's of tool radius compensation

- 刀具半径补偿模式设定 G138/G139/G237/G238/G239/G05
- 刀具半径补偿路径 G25\G26

Example 1	G138	G139	G05
G25	Path 1	Path 2	Path 3

```
*WZKG25 (Contour for G25)
N10 G00 G90 T1 D1 X0 Y0 Z0 G17
```

(Presentation of contour)

```
N15 G01 X20 Y20 F1000
N20 G91
N25 G1 X10
N30 X5 Y-5
N35 Y-5
N40 X-5 Y-3
N45 G01 G90 X0 Y0 F1000
```

(Path 1)

```
N100 G138 G41 (Direct selection and TRC left of contour)
N105 G01 X20 Y20 F1000 (Compensation movement after G41)
N110 G25 (G25 linear transition)
N115 G1 G91 X10
N120 X5 Y-5
N125 Y-5
N130 X-5 Y-3
N135 G138 G40 (Direct deselection and TRC-off)
N140 G01 G90 X0 Y0 F1000 (Compensation movement after G40)
```

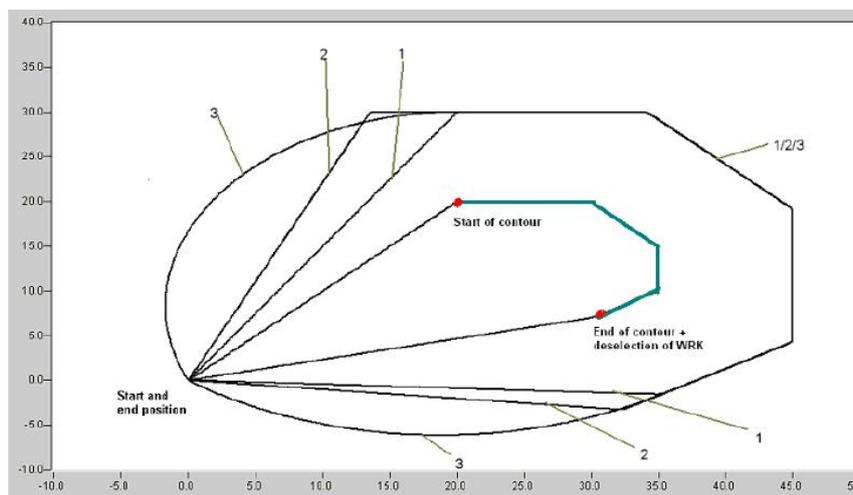
(Path 2)

```
N200 G139 G41 (Indirect selection and TRC left of contour)
N205 G01 X20 Y20 F1000 (Compensation movement after G41)
N210 G25 (G25 linear transition)
N215 G1 G91 X10
N220 X5 Y-5
N225 Y-5
N230 X-5 Y-3
N235 G139 G40 (Indirect deselection and TRC-off)
N240 G01 G90 X0 Y0 F1000 (Compensation movement after G40)
```

(Path 3)

```
N300 G05 G41 (Tangential selection and TRC left of contour)
N305 G01 X20 Y20 F1000 (Compensation movement after G41)
N310 G25 (G25 linear transition)
N315 G1 G91 X10
N320 X5 Y-5
N325 Y-5
N330 X-5 Y-3
N335 G05 G40 (Tangential deselection and TRC-off)
N340 G01 X20 Y20 F1000 (Compensation movement after G41)
```

```
N999 M30
```



G26 设定下，各种补偿模式案例：

Example 2	G138	G139	G05
G26	<i>Path 4</i>	<i>Path 5</i>	<i>Path 6</i>

```
%WZKG26 (Contour for G26)
N10 G00 G90 T1 D1 X0 Y0 Z0 G17
```

(Presentation of contour)

```
N15 G01 X20 Y20 F1000
N20 G91
N25 G1 X10
N30 X5 Y-5
N35 Y-5
N40 X-5 Y-3
N45 G01 G90 X0 Y0 F1000
```

(Path 4)

```
N400 G138 G41 (Direct selection and TRC left of contour)
N405 G01 X20 Y20 F1000 (Compensation movement after G41)
N410 G26 (G26 circular transition)
N415 G1 G91 X10
N420 X5 Y-5
N425 Y-5
N430 X-5 Y-3
N435 G138 G40 (Direct deselection and TRC-off)
N440 G01 G90 X0 Y0 F1000 (Compensation movement after G40)
```

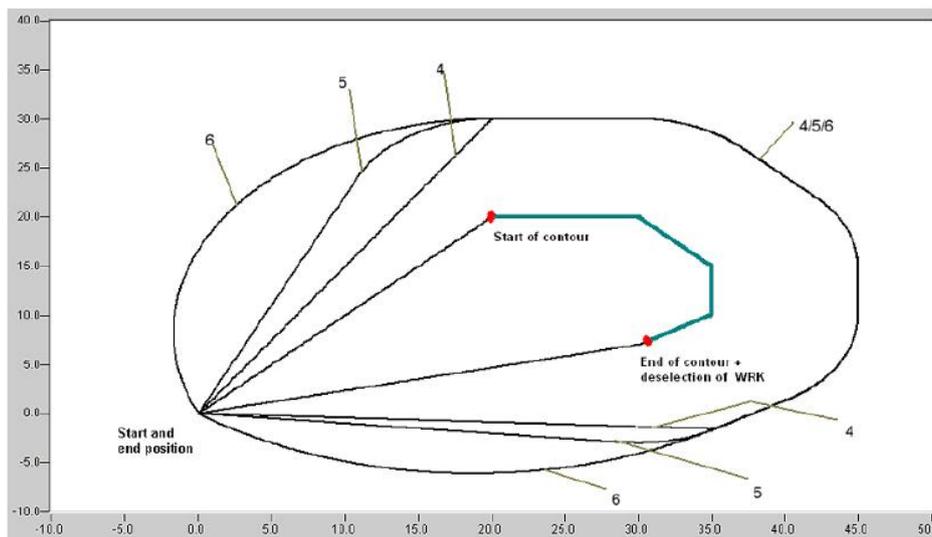
(Path 5)

```
N500 G139 G41 (Indirect selection and TRC left of contour)
N505 G01 X20 Y20 F1000 (Compensation movement after G41)
N510 G26 (G26 circular transition)
N515 G1 G91 X10
N520 X5 Y-5
N525 Y-5
N530 X-5 Y-3
N535 G139 G40 (Indirect deselection and TRC-off)
N540 G01 G90 X0 Y0 F1000 (Compensation movement after G40)
```

(Path 6)

```
N600 G05 G41 (Tangential selection and TRC left of contour)
N605 G01 X20 Y20 F1000 (Compensation movement after G41)
N610 G26 (G26 circular transition)
N615 G1 G91X10
N620 X5 Y-5
N625 Y-5
N630 X-5 Y-3
N635 G05 G40 (Tangential deselection and TRC-off)
N640 G01 X20 Y20 F1000 (Compensation movement after G41)
```

```
N999 M30
```



3、前馈控制功能 G135/G136/G137

3.1 指令格式

G135

G136 X_ Y_

G137

3.2 指令说明

G135 激活前馈控制功能，G136 设定各轴对应的前馈控制百分比，G137 取消前馈功能。

G136 前馈控制设定百分比限制小于 100%

3.3 指令举例



Programming example

```
G135      (Selection of feedforward control: weighting for all axes 100%)
```

```
G136 X80 Y95 Z0 (Weighting; here Z-axis receives no feedforward control)
```

4、快速移动速度权重 G129

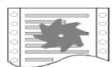
4.1 指令格式

G129= <VAR>

4.2 指令说明

通过 G129 功能可以改变 G00 速度；

4.3 指令举例



Programming example

```
N10 G129 = 70      (Axis group-specific rapid traverse weighting)
                   (Rapid traverse velocity of all axes to 70%)
```

```
N20 G00 X100 Y150 (Linear interp., rapid traverse movement with 70%)
```

Special feature:

```
N50 G129 = 70 X10 Y20 (In the same block with G129 also axis)
                       (positions can be programmed)
```

5、加速度权重 G130/G131

5.1 指令格式

G130 X_ Y_

G131=<VAR>

5.2 指令说明

通过 G130/G131 指令可以改变加速度。

5.3 指令举例



Programming example

```
N10 G130 X70      (Axis-specific acceleration weighting:)
                   (Acceleration of X-axis is restricted to 70%)
```

```
N20 G01 F1000 X100 (Linear interpolation)
```

```
N30 G130 Y60      (Acceleration of Y-axis is restricted to 60%)
```

```
N40 Y100          (Acceleration of X-axis remains set to 70%)
```

```
N40 Y100          (Linear interpolation)
```

```
N50 G131 = 100    (Axis group-specific acceleration weighting:)
                   (Acceleration of all axes to 100%)
```

Special feature:

```
N50 G131 = 100 X10 Y20 (In the same block with G131 also axis)
                       (positions can be programmed)
```

6、Jerk 上升时间 G132\G133\G134

6.1 指令格式

G132 X<VAR>

G133=<VAR>

G134=<VAR>

6.2 指令说明

通过 G132\G133\G134 可以修改 jerk 上升时间;

6.3 指令举例



Programming example

```

N10 G132 X200      (Axis-specific ramp time weighting:)
                  (Ramping time of X-axis is increased by 200 %)
N20 G01 F1000 X100 (Linear interpolation)
N30 G132 Y50      (Ramping time of Y-axis is decreased by 50%)
                  (Ramping time of X-axis remains at 200%)
N40 Y100         (Linear interpolation)
N50 G133 = 100    (Axis group-specific acceleration time weighting:)
                  (Ramping time of all axes to 100%)
N60 G134 = 50     (Axis group-specific acceleration time weighting:)
                  (Geometrical ramping time of all axes to 50%)

Special feature:
N50 G133 = 100 X10 Y20 (In the same block with G133 also axis positions)
                       (can be programmed)
N60 G134 = 50 X10 Y20 (In the same block with G134 also axis positions)
                       (can be programmed)

```

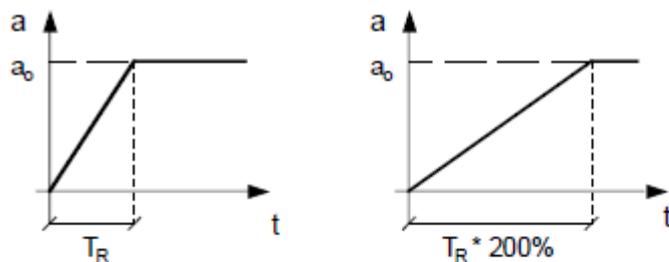


Fig. 4.52: Example for weighting of ramp time with G132/G133

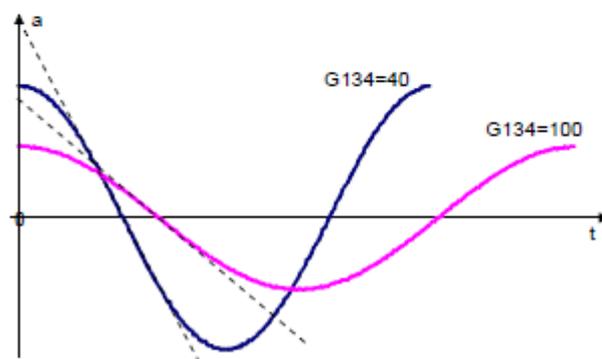


Fig. 4.53: Example for weighting of ramp time with G134 at circular interpolation

7、测量功能 G100/G310/G101/G102

测量模式定义

测量模式共包括 7 中不同的模式，如下：

Measuring types	Significance
1	Measurement travel with at least one axis, Measurement feed programmable through F-Word.
2	Measurement travel with exactly one axis. Measurement feed is specified in the axis parameter list [2]-5.
3	Measurement travel with at least one axis, Measurement feed programmable through F-Word , optionally further traverse up to the destination point.
4	Measurement travel only with the maximum of 3 main axes, Measurement feed programmable through F-Word.
5	Interruptible measurement travel with at least one axis, Measurement feed programmable through F-Word.
6	Interruptible measurement travel with at least one SERCOS-Axis, Measurement feed programmable through F-Word.
7	Measurement travel with moving on a fixed stop with at least one axis, Measurement feed programmable through F-Word

指令举例：



Programming example

```
%Meas_traverse
N10 G00 X0 Y0 Z0
N20 X5
N30 G100 X10 Y10 F500
N40 G01 X7
N50 M30
```

Fig. 4.6 shows the representation of the resulting path:

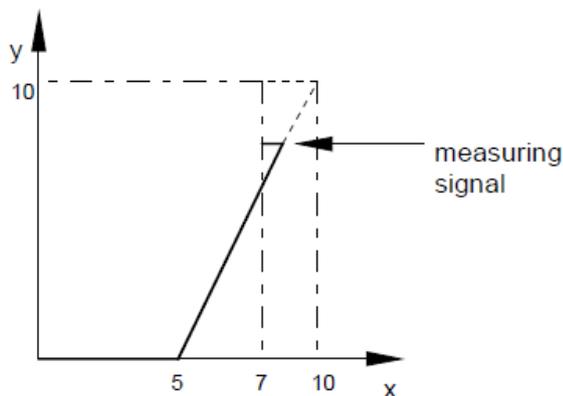


Fig. 4.6: Programmed path

详细功能介绍参照【[第六章—功能七---测量功能](#)】

8、G 代码子程序 G80-G89\G800-G819

两种方式定义子程序

- 1、配置文件中，直接定义 G80 对应的子程序名称；
- 2、通过#FILENAME 方式，定义子程序对应；



Programming examples

```
%example1
N10 #FILENAME[ MSG="prog_flow.txt" ]
N20 $IF V.E.PLC_START_HOME == 1
N30 G74 X1 Y2 Z3
N40 #MSG SAVE["Homing executed"]   Output in prog_flow.txt
N50 $ENDIF
:
Nxx #MSG SAVE["Roughing OK"]
:
Nxx #MSG SAVE["Finishing OK"]
N985 V.E.WP_CNTR = V.E.WP_CNTR+1
N990 #MSG SAVE["Workpiece No. %d OK", V.E.WP_CNTR]
M1000 M30

%example2
N10 #FILENAME[ M6="tool_change.nc" ]
N20 G00 X100 Y100 Z0
N30 M6           Call of tool change program tool_change.nc
:
M1000 M30

%example3
N10 #FILENAME[ G80="g80_up_test.nc" ]
N20 G00 X100 Y100 Z50
N30 G80           Call of sub program g80_up_test.nc
:
M1000 M30

%example4
N10 #FILENAME[ G800="g800_up_test.nc" G815="g815_up_test.nc"]
N20 G00 X100 Y100 Z50
N30 G800           Call of sub program g800_up_test.nc
:
N90 G815           Call of sub program g815_up_test.nc
:
M1000 M30
```

详细功能介绍参照 [【第六章—功能八---子程序功能】](#)

七、G 代码循环语句

- 1、\$IF指令
- 2、\$SWITCH
- 3、\$GOTO
- 4、\$FOR
- 5、\$WHILE
- 6、\$DO
- 7、\$REPEAT
- 8、\$BREAK
- 9、\$CONTINUE

详细功能介绍参照 [【第六章—功能十---NC代码循环指令】](#)

八、辅助功能 M 函数

特殊 M 函数:

M00	Programmed stop 179
M01	Optional stop 179
M02	Program end 179
M03	Spindle rotation cw. 434, 467
M04	Spindle rotation ccw. 434, 467
M05	Spindle stop 434, 467
M17	End of subroutine 179
M19	Positioning of spindle 435, 469
M29	End of subroutine 179
M30	Program end 179
M40- 45	Selection of spindle gear ranges 440

九、特殊功能符

D	Tool geometry compensation 368
E*	Feedrate at block end 184
F	Feedrate at block start 184
H	<i>Technological information to PLC 178, 472</i>
L	Subroutine call (global) 188
L	CYCLE Cycle call 193
LL	Subroutine call (local) 186
N	Block number 185
P	Parameter calculation 197
R	Radius programming 141
S	Spindle speed 437, 471
S.POS	Spindle position 435, 469
T	Tool selection 183
/	Skipping of NC-blocks 21

十、附加功能

#ACHSE	Programming of axes for manual operation 163
#ACS ON / OFF	Definition/activation of a fixture adaptive coordinate system 514
#ADD	Additional information at block end 277
#AX DEF	Definition of an axes configuration (extended syntax) 243
#AX DEF DEFAULT	Loading of default axes configuration (extended syntax) 243
#AX LINK ON / OFF / ALL	<i>Programming coupling instructions (extended syntax)</i> 261, 268
#AX REQUEST	Requesting of axes (extended syntax) 234
#AX RELEASE	Releasing of axes (extended syntax) 241
#AX RELEASE ALL	Releasing of all axes (extended syntax) 241
#BACKWARD STORAGE CLEAR	Clearing of backward storage 362
#BLOCKSEARCH LOCKED/RELEASED	Locking of program areas for block search 364
#CALL AX	Requesting of axes 224
#CAX	Requesting of a spindle axis for C-axis operation 454
#CAX OFF	Conversion of C-axis into spindle 454
#CAXTRACK ON/OFF	Automatic axis tracking 334
#CHANNEL INIT	<i>Acceptance of current command or actual positions of all or some specific axes</i> 170, 171
#CLEAR CONFIG	Deleting of a saved configuration 356
#COMMAND WR / SYN	Writing of SERCOS-commandos 303, 304
#COMMAND WAIT / SYN	Waiting for SERCOS-commandos 305, 306
#COMMENT BEGIN / END	Inter-block valid comments 250
#CONTOUR MODE	Parameterizing of polynomial contouring 106
#CONTROL AREA START / END	Definition of control areas 358
#CONTROL AREA ON / OFF	Selection/deselection of control areas 360
#CONTROL AREA CLEAR	Deleting of control areas 360
#CS ON/ OFF	Definition/activation of a machining coordinate system 509
#CYL	Selection of lateral surface machining 460
#CYL OFF	Deselection of lateral surface machining 461
#DELETE	Deleting of self-defined variables or parameters 421
#DRIVE WR SYN / WAIT SYN	Switching of drive functions 366
#ECS ON / OFF	Selection/deselection of effector coordinate system 521
#ENABLE AX LINK / #DISABLE AX LINK	Selection of coupling instruction 268
#ERROR	User-defined error output 338
#EXPL SYN	Explicite synchronisation 484
#FACE	Selection of facing 457
#FACE OFF	Deselection of facing 458
#FGROUP	Definition of a feed group 343
#FGROUP ROT	Feed axes for rotary workpieces 343
#FGROUP WAXIS	Weakest axis becomes feed axes 343
#FILENAME	Definition of file names 352
#FILTER	Filter programming 298
#FLUSH	Flushing CNC-channel with interruption of movement 248

#FLUSH CONTINUE	Flushing CNC-channel without stopping of movement 248
#FLUSH WAIT	Synchronization of decoding and interpolation 249
#FREE TOOL CHANGE ON / OFF	Tool change during active synchronous operation 362
#GET IPO OFFSET	Acceptance of current offsets 169
#GET WCS POSLIMIT	Motion limits within a workpeace coordinate system 528
#HSC ON / OFF	Free form surface programming 290
#IDENT WR / RD / SYN	Writing/Reading of SERCOS- parameters 300
#IF / ELSE / ENDIF	Conditional interpretation 254
#INIT MAKRO TAB	Initialization of macro table 498
#KIN ID	Machine kinematics 506
#LOAD CONFIG	Loading of a saved configuration 355
#MACHINE DATA	Writing of machine data 349
#MAIN SPINDLE	Switching of main spindle 479
#MCS ON / OFF	Temporary transition to the machine axes coordinate system523
#MCS TO WCS	Transformation of machine coordinate into workpiece coordinate526

十一：变量设定

V.A.	Axis-specific variables
V.SPDL.	Spindle-specific variables
V.G.	Global variables
V.P.	Self-defined variables, program global
V.S.	Self-defined variables, global
V.L.	Self-defined variables, program local
V.E.	External variables
V.TOOL.	Tool identification variables
V.TLM.	Tool life variables

详细功能介绍参照 [【第六章—功能九---变量】](#)

十二、PLC OPEN 功能

MC_Home	Reference point travel 488
MC_MoveAbsolute	Movement of the axis to an absolute position 489
MC_MoveAdditive	Relative motion in addition to the commanded position 490
MC_MoveRelative	Relative motion in addition to the current 491
MC_MoveSuperImposed	Relative motion in addition to a motion already active 492
MC_MoveVelocity	Endless motion at the specified speed 493
MC_Stop	Stopping of an axis motion 494
MC_GearIn	Gear coupling with a gear ratio 495
MC_GearOut	Releasing of a gear coupling 496
MC_Phasing	Phase offset of couplings 497

十三、附加的轴功能

INDP_SYN	Synchronous (block wise) independent axis movement 540
INDP_ASYN	Asynchronous (multi block) independent axis movement 540
OSC	Oscillating axes 544
COMP	Selection/deselection of axis compensations 549
DIST_CTRL	Distance controlled axes 550
OVERRIDE	Programmable axis override 551
DYNAMIC	Programmable acceleration overload 552

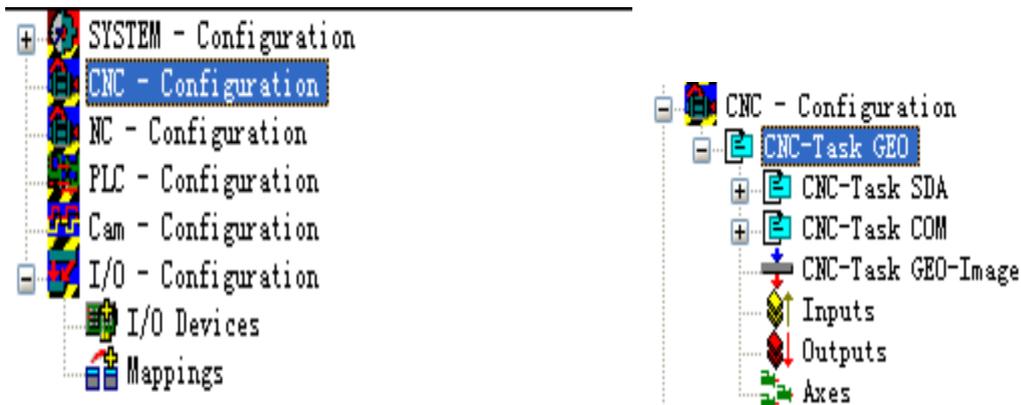
第四章 TwinCAT CNC 配置文件

概述:

本章旨在介绍如何配置如何新建一个 CNC 工程组态配置文件，并对配置文件中的信息进行讲解。CNC 程序包括 TwinCAT CNC 的配置文件、CNC 框架 PLC 程序和界面组成。

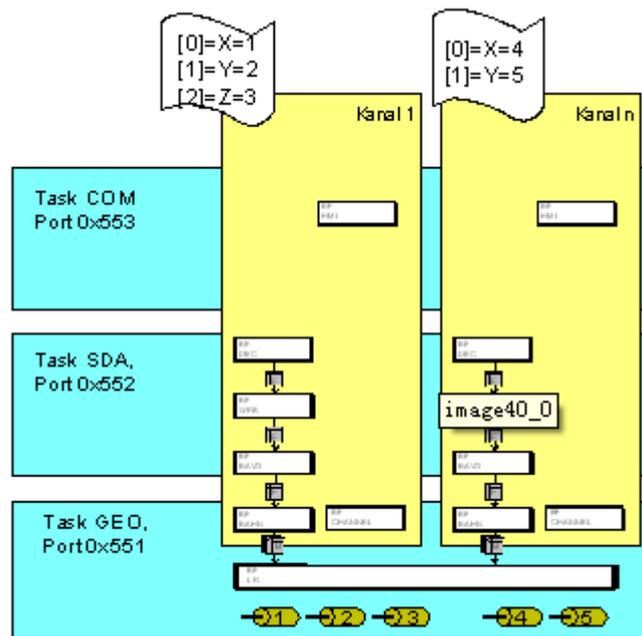
CNC 配置介绍:

CNC 配置文件与 NC 工程文件的区别在于多了个 CNC 配置项，所有的 CNC 功能的配置都在这里进行。包括对于 CNC 的插补周期、路径规划、HMI 与 CNC 通讯、通道参数、轴参数配置等



在 CNC_Configuration 处右击，添加一个 CNC 任务，包括 Task GEO、SDA、COM 等，Task GEO 对应的接口为 PORT 0X551，SDA 对应接口为 PORT 0X552，TASK COM 对应接口为 PORT 0X553。

访问各个接口的参数需要通过 ADS 通讯的方式，通过 ADS-NetId 地址，index Group 以及 index Offset。



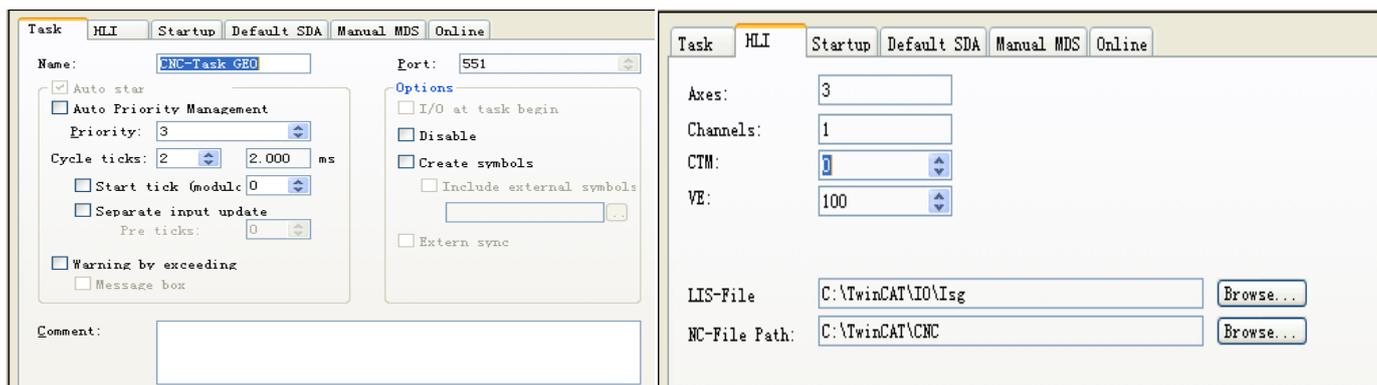
一、CNC-Task GEO

1、TASK

- 1) CNC-Task GEO 通讯端口 551;
- 2) 这个 Task 要设置为最高优先级;
- 3) 设置 CNC 的插补周期;
- 4) 根据相应的插补曲线生成插补点。

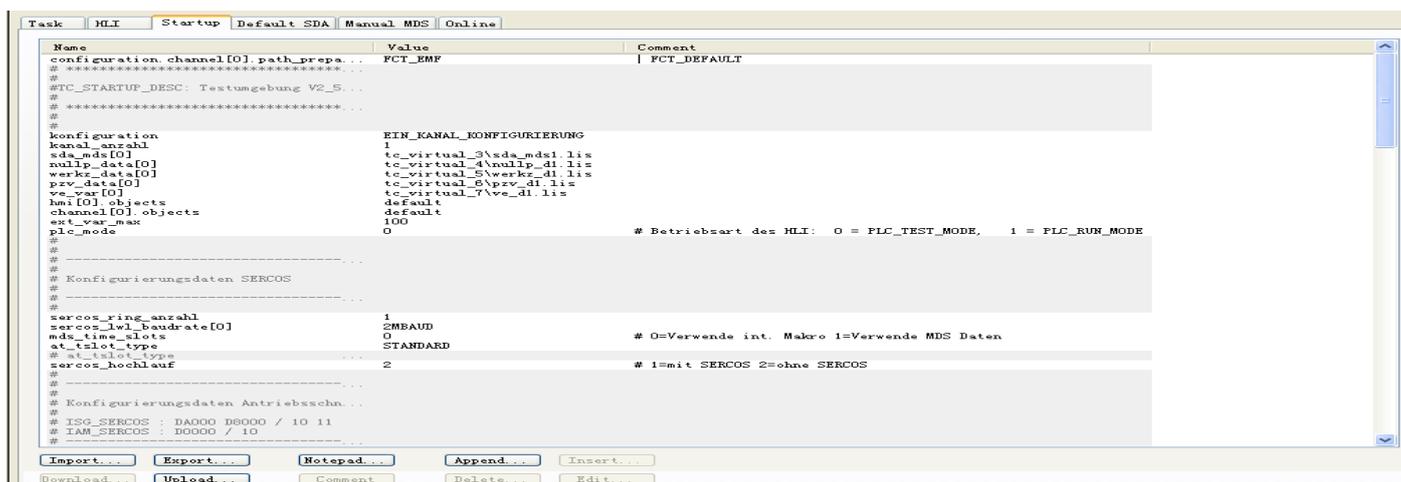
2、HLI

- 1) HLI=high level interface;
- 2) HLI 负责 CNC 通道与 PLC 之间的通讯，是一块共享内存区;
- 3) HLI 被划分为全局、通道区域、轴区域并包括 VE 全局变量以及 CTM (Channel Task Manager);
- 4) 这里的轴数量以及通道数量是自动赋值的，而 CTM 以及 VE 变量部分则是部分手动赋值。
- 5) 共享内存区 HLI 在 CNC 通道初始化生成，开始执行 PLC 程序时，即自动通过 PLC 库通过指针指向相应的内存区位置。



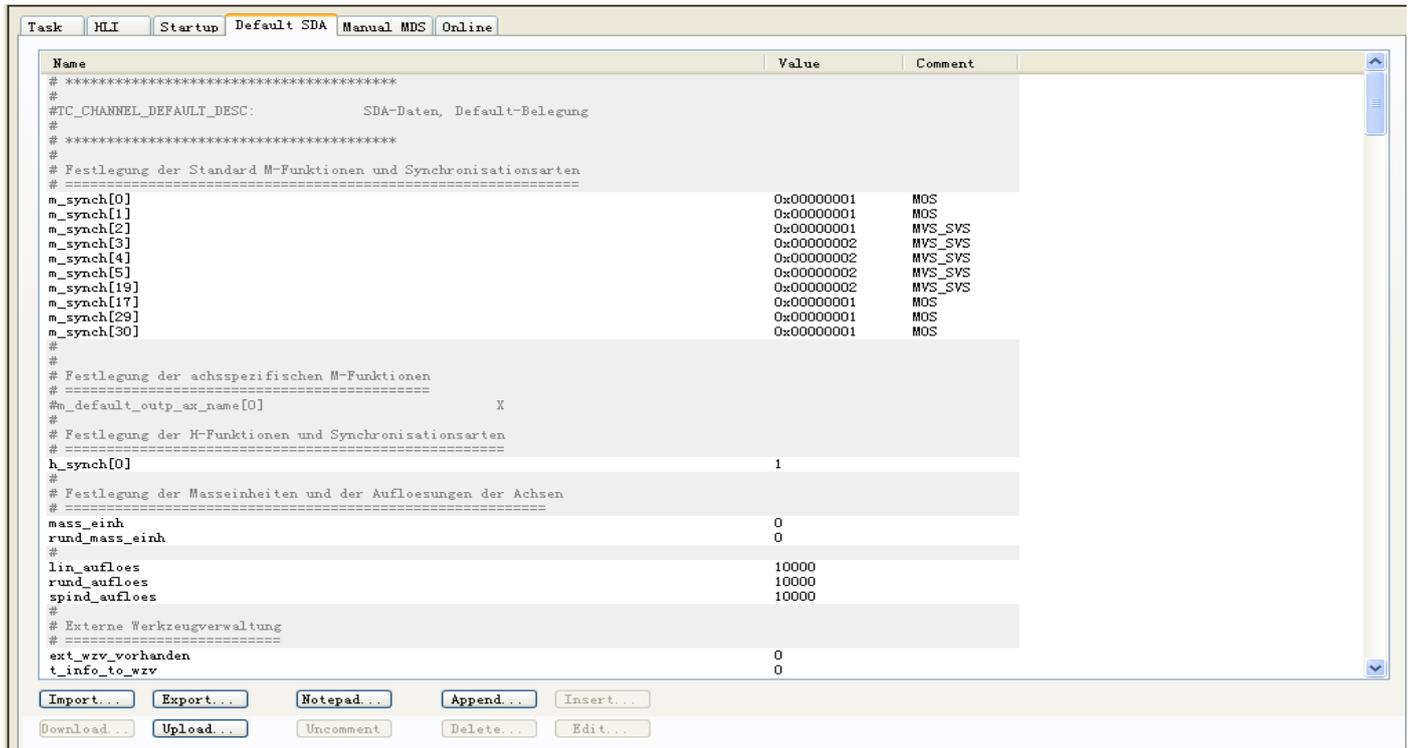
3、Startup

- 1) 注释必须包括 '#' 字符，并跟随一个空格;
- 2) 该文件包含的是应用数据设置以及 NC 启动表格设置;
- 3) 包含启动的通道数、轴数、sercos 驱动应用、刀具表格、坐标系偏置表格、夹具偏置表格、手动操作表格、回退参数设置、外部变量数量设置。



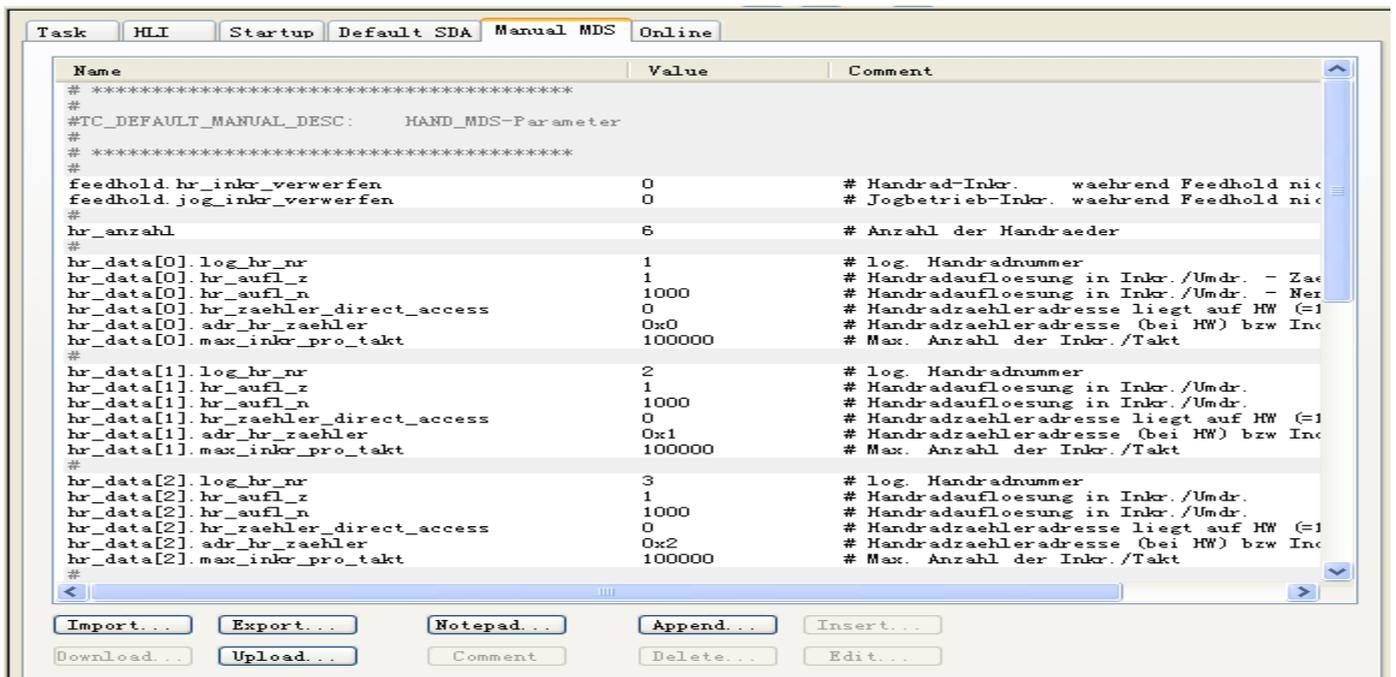
4、Default SDA

默认激活的 SDA，与通道中的 SDA 重合，优先使用通道中的 SDA 表格；



5、Manual MDS

手动设置，主要包含的是手摇轮部分的设置。

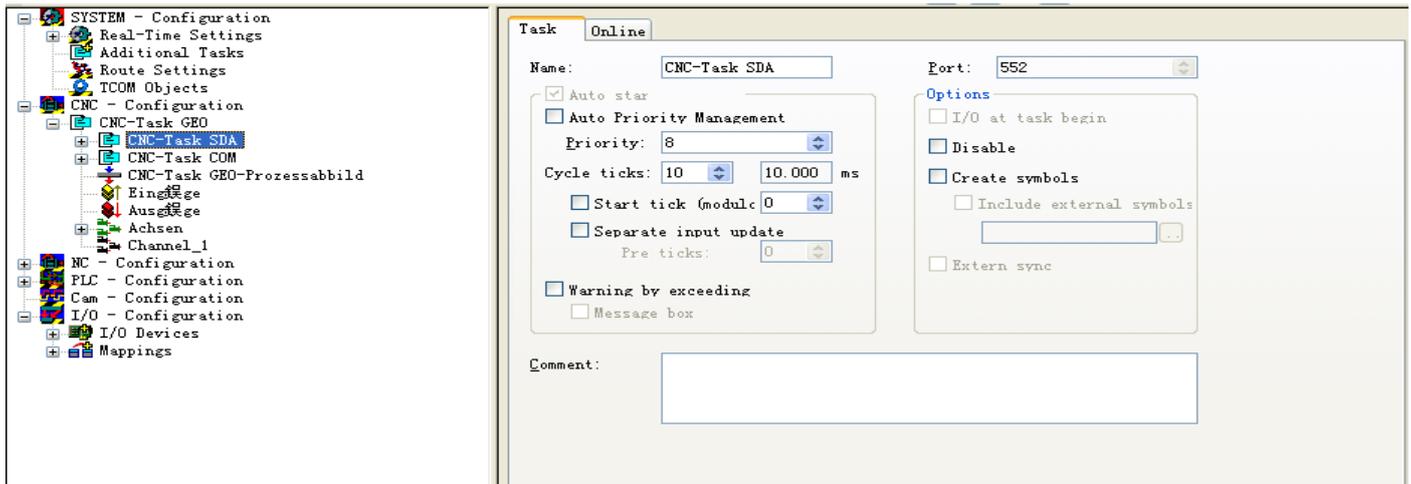


6、ONLINE

显示当前的处理器负载状况。

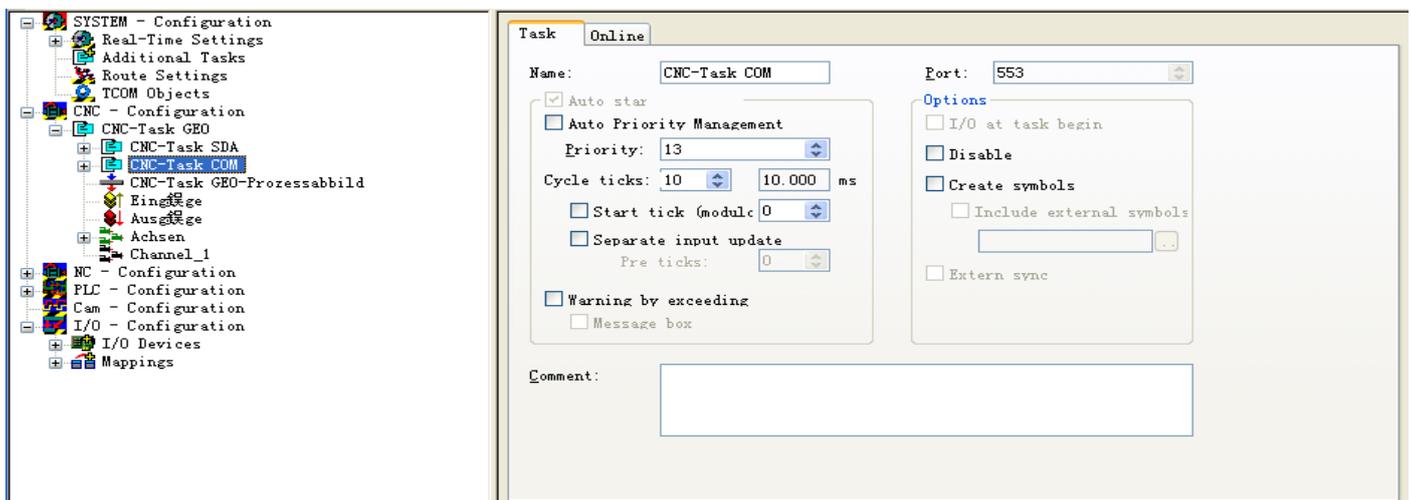
二、CNC-Task SDA

- 1、CNC-Task SDA 的通讯端口为 552;
- 2、对 NC 程序预读，路径规划;
- 3、优先级低于 CNC-Task GEO。



三、CNC-Task COM

- 1、CNC-Task COM 通讯端口为 553;
- 2、处理 CNC 通道与外部接口通讯;
- 3、这个 TASK 管理所有的 ADS 通讯。



四、Channel 参数

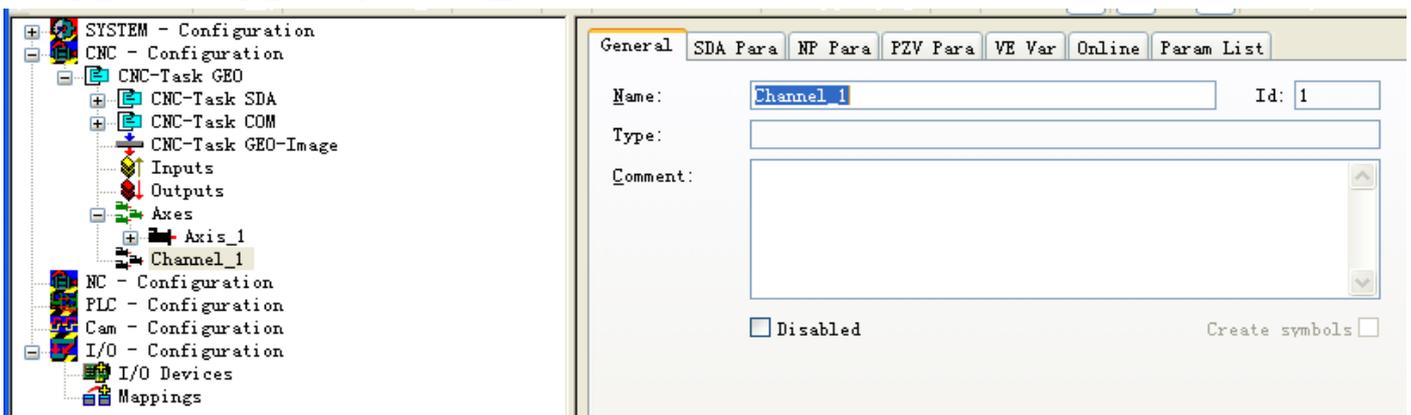
概述:

在一个 CNC 工程中，需要包括一个通道以及一个 CNC 轴。添加一个通道要在“GEO CNC TASK”的基础上，作为一个组成部分。在 CNC-Task GEO 处右击，添加一个通道 channel_1（若为多通道，则可在该处添加多个通道，最多 12 通道），通道参数设置、坐标系偏置设置、VE 变量设置、螺距补偿、外部刀具参数表格，并可以在通道中进行手动、自动、半自动简单操作。

- CNC 中最多可以添加 12 个通道；
- 每个通道都是独立的，可以独立运行 CNC 程序；
- 每个通道中最多可以执行 32 根 CNC 轴。
- 前面在 startup 中设置 default SDA 在此处被当前的 SDA 覆盖，如果在 channel 中的 SDA 没有设置或者设置相同则 Default SDA 在此处仍然有效；

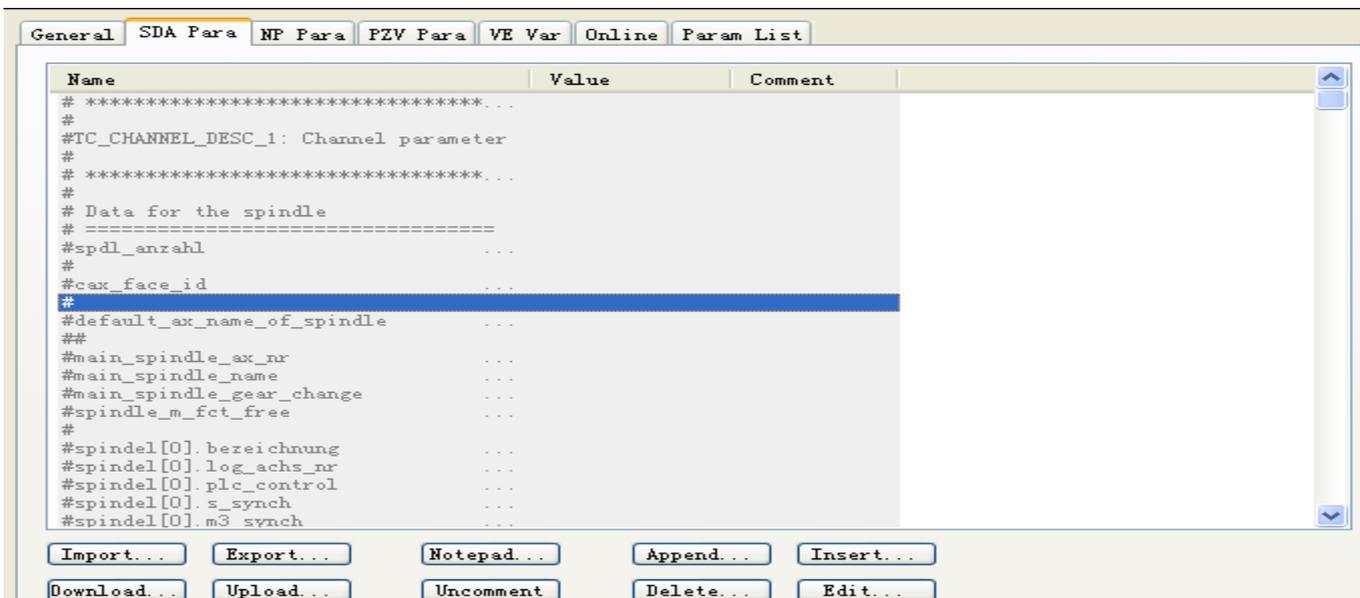
1、General

通道名称以及通道的系统 ID 在此设置；



2、SDA Para 参数

SDA Para 参数主要对于通道中的主轴、M\H 函数、子程序、刀具补偿等进行了设置。



- 主轴设置

轴参数设置为 spindle 轴;

Spdl_anzahl: 通道中包含的 spindle 轴数量;

default_ax_name_of_spindle: 默认的主轴名称;

main_spindle_ax_nr : 主 spindle 轴逻辑轴号, 当在 NC 代码#MAIN SPINDLE 认为调用的该轴;

main_spindle_name: 主 spindle 轴名称;

main_spindel_gear_change : 主轴齿轮比是否生效;

spindel[0].bezeichnung S1: 逻辑轴名称必须以字母 S 打头,

spindel[0].log_achs_nr 6 : 逻辑轴号, 同时逻辑轴号不能够分配多次, 同时一个逻辑轴号不能够同时分配为轨迹轴并且分配为主轴;

plc_control : 主轴控制类型选择;

s_synch : 主轴速度设定同步类型;

m3_synch : 正向旋转的同步类型;

m4_synch : 逆向旋转的同步类型

m5_synch : 停止旋转的同步类型

m19_synch : 位置控制的同步类型

spindle_m_fct_free: 是否使用 M 函数;

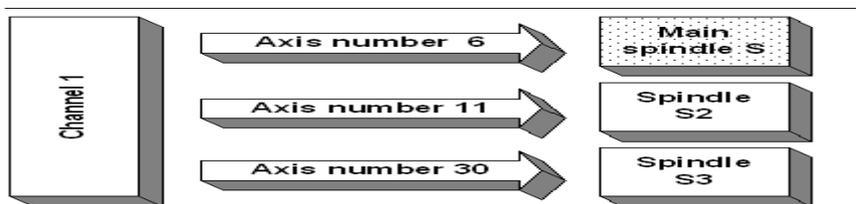


Fig. 2-5: Spindle configuration example

- 法相跟随轴 Tracking of a rotator axis

log_number_tracking_axis 5: 法相跟随轴轴号;

auto_align_tracking_axis: 自动跟随是否有偏置;

fast_tracking_transition: 法相跟随是否停止进行调整;

suppress_cs_tracking_offset: 是否使用旋转坐标系系统;

tracking_axis_rot_wc : 刀具、工件跟随选择;

- 初始执行 G 代码

start_init_prog_file: 程序执行时, 首先执行的是这个参数对应的 G 代码;

注意: 在半自动和手动模式下, 这段程序不执行;

- 执行 M6 代码

m6_prog_file tool_change1.nc

当 m6_prog_file 在通道参数中设置值时, 这个 M 函数就变成一个特殊函数类似于一个子程序, 直接盗用该参数中对应的 G 代码。如果没有设置任何至, 那么它仍然是如普通的 M 函数使用。

#FILENAME[M6=" tool_change1.nc"]

- G 代码子程序

g80_prog_file G80_up_test.nc (Name of sub program for G80)

1) 在轴参数中设置变量如上, 等同于在 NC 代码程序中使用#FILENAME[G80=" G80_up_test.nc"];

2) G80 类似子程序包括 (G80-G89 共 10 个);

3) 当 g80_prog_file 在通道参数中设置为 G80_up_test.nc 的时候, 在 NC 代码中调用 G80 时, 即执行的是 G80_up_test.nc 程序。

4) G800-G819 也是与 G80 类似的指令。

- M 函数

M_synch[i] 0X0000002 MVS_SVS

1) M 函数个数为 0-65535 个;

2) M 函数的类型包括 NO_SYNCH, MOS, MVS_SVS, MVS_SNS, MNS_SNS, MNE_SNS, MVS_SLM, MVS_SLP, MOS_TS, MEP_SVS, MET_SVS, BWD_SYNCH, FWD_SYNCH, FAW_SYNCH 等。触发顺序不同。

- H 函数

H_synch[i] 0X00000002 MVS_SVS

1) H 函数个嘴为 0-65535 个;

2) H 函数的类型包括 NO_SYNCH, MOS, MVS_SVS, MVS_SNS, MNS_SNS, MNE_SNS, MEP_SVS, MET_SVS, BWD_SYNCH, FWD_SYNCH, FAW_SYNCH 等。

- 单位转换

mass_einh 0

单位使用, 0 为米制, 1 为英尺单位。默认为米制 0;

rund_mass_einh 0

单位使用, 0 为 degree, 1 为 new degree 单位。默认为 degree 0;

lin_aufloes 10000

单位使用, 直线运动, 以 0.1um 为单位, 直线运动在 NC 代码中以 1mm 为单位

rund_aufloes 10000

单位使用, 旋转轴运动, 以 0.0001° 为单位, 旋转轴运动在 NC 代码中以 1° 为单位

spind_aufloes 10000

单位使用, 主轴运动, 以 0.0001° 为单位, 主轴在 NC 代码中以 1° 为单位。

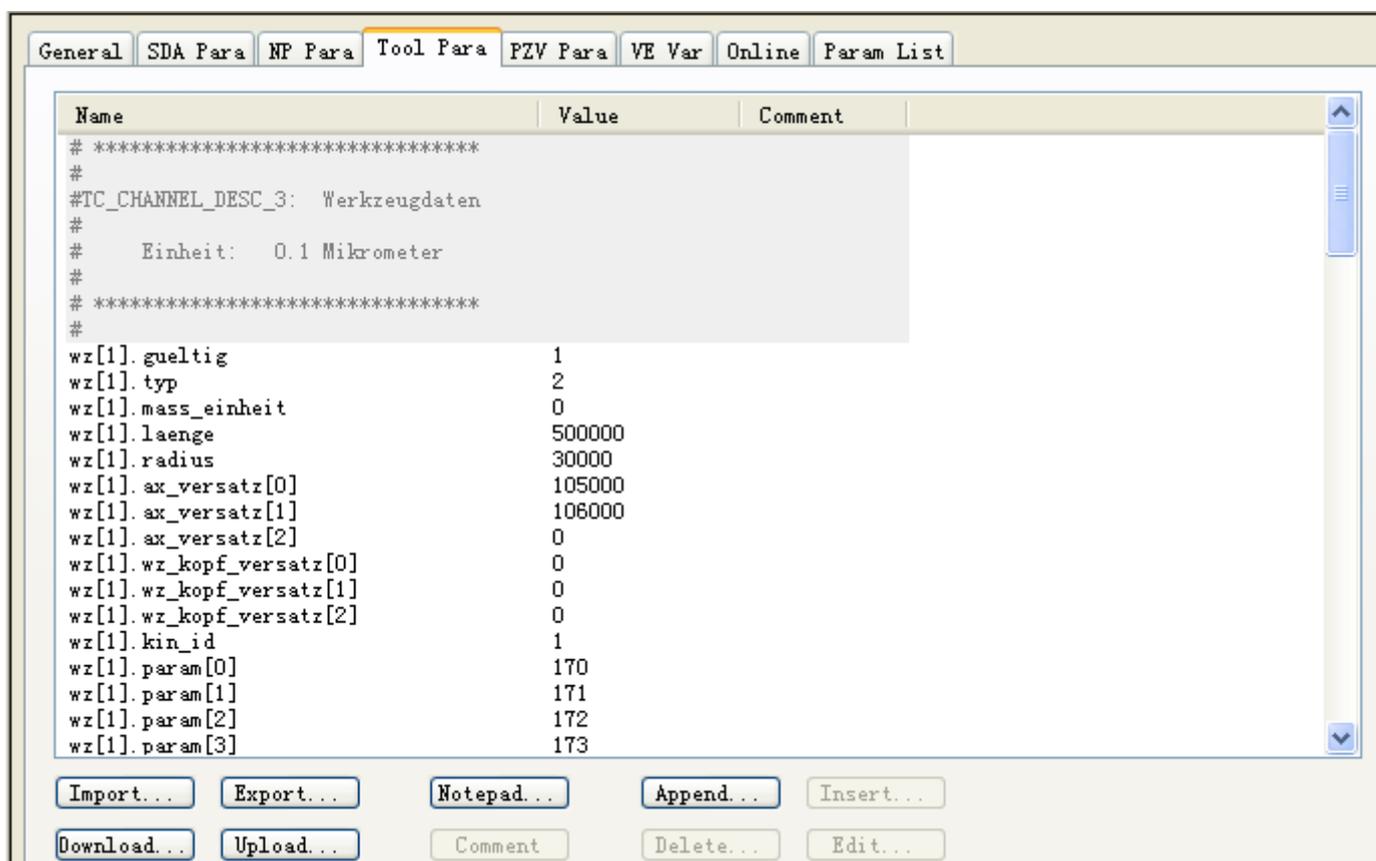
- 刀具管理

einrechnen_mit_t 1

当 einrechnen_mit_t 设置为 1 时, 在程序中执行 T80 的是时候, 相应的 T80 D80 都同时执行。

ext_wzv_vorhanden 1

外部刀具管理表格, 当 ext_wzv_vorhanden 设为 0 时, 出现外部刀具管理表格。如下图。



- 测量功能

测量功能要配合 NC 代码以及 PLC 程序对于触发信号进行处理。

Seven different measurement types are available. With this element, the desired measurement type can be set.

Variable name	Type	Permitted range	Dimension
messtyp	UNS16	1, 2, 3, 4, 5, 6, 7	----

Value	Significance
1	Measurement travel with at least one axis, Measurement feed programmable through F-Word.
2	Measurement travel with exactly one axis. Measurement feed is specified in the axis data list.
3	Measurement travel with at least one axis, Measurement feed programmable through F-Word, optionally further traverse up to the destination point.
4	Measurement travel only with the maximum of 3 main axes, Measurement feed programmable through F-Word.
5	Interruptible measurement travel with at least one axis, Measurement feed programmable through F-Word.
6	Interruptible measurement travel with at least one SERCOS-Axis, Measurement feed programmable through F-Word.
7	Measurement travel by moving on a fixed stop with at least one axis, Measurement feed programmable through F-Word.



The different measurement types 1-6 are also described in [\[PROG\]](#).

EXAMPLE: Selection of measurement type 3 for a measurement travel with two axes and subsequent further traverse up to the programmed destination point.

Extract from channel parameters list:

```
# Setting of measurement type
messtyp 3
```

- 校验功能

mittelpkt_diff 1000

设置最小圆的精度，允许的绝对中心点偏移为 0.1mm。

mittelpkt_faktor 5

允许的相对中心点偏置为 0.5%

max_radius_diff_circle 100

圆最大允许的起点与终点差异最大为 0.01mm。

max_proz_radius_diff_circle 10

圆最大允许的半径最大差异是 1%。

- 单步执行

einzelstschrittmodus 0

单步执行-1 为仅仅在 NC 运动块中才可执行单步模式；0 在 NC 运动块和相关控制块时执行；1 在所有运动会块中执行单步模式。

- 通道倍率限制

max_vb_override 1500

通道最大的倍率设置为 150%。

override_delay 50000

为避免在改变倍率时，导致机械震荡，延时执行倍率设定 50ms（单位 1us）

- 复位轴状态设定

reset_no_axis_to_axv 1

当复位时，0 为所有的轴单独控制，1 为所有轴仍然在通道中。

- G90/G91 执行

reset_no_axis_to_axv 1

0: 在同一个 NC 程序中只能够执行一个；1: 可以在同一个 NC 程序中执行 G90 G91，直到只想到下一个 G90/G91

- 速度、加速度、加速度限制

```
vector.velocity      1500      mm/min
vector.acceleration  1800000  mm/min~2
vector.deceleration  2000000  mm/min~2
```

速度、加速度、减速度不超过上述设置。

- 速度前馈限制

```
# Parametrierung Speed limit Look Ahead
```

```
# =====
```

```
#speed_limit_look_ahead.f_enable      1      0: disabled 1: enabled
#speed_limit_look_ahead.v_limit      750    [0.1%] of programmed speed
#speed_limit_look_ahead.f_time      1 unit of following parameters time or path 0: path 1: time
#speed_limit_look_ahead.dist_to_corner      50000  [0.1um or us]
#speed_limit_look_ahead.dist_from_corner      60000  [0.1um or us]
speed_limit_look_ahead.f_override_weight_v_limit      0      0: disabled 1: enabled
```

1) 该参数可以通过 NC 代码来设定 V.G.SPEED_LIMIT.*;

2) f_enable 设定是否使用进给前馈功能; v_limit 设定在前馈检测到速度限位时执行限位的速度值, 此值的设定值: $V=F \text{ 速度} * v_limit / 1000$; f_time 设定 0: dist_to_corner / dist_from_corner 设置为距离, 1: 设定为时间; dist_to_corner : 设定在到达拐角前起作用的时间或者距离; dist_from_corner: 设定在拐角之后起作用的时间或者距离; f_override_weight_v_limit: 设定倍率是否对拐角速度有效。

- 启动 G 代码指令

```
# Default values for G-Functions
```

```
prog_start.g_gruppe[0].nr 0 (Area: WEG_BED)
prog_start.g_gruppe[4].nr 17 (Area: EBENE)
prog_start.g_gruppe[6].nr 25 (Area: WRK_UEBER)
prog_start.g_gruppe[8].nr 52 (Area: DURCHM)
prog_start.g_gruppe[12].nr 71 (Area: MASSEINH)
prog_start.g_gruppe[14].nr 90 (Area: MASSANG)
prog_start.g_gruppe[21].nr 135 (Area: VORST)
prog_start.g_gruppe[22].nr 139 (Area: WRK_E_A)
prog_start.g_gruppe[23].nr 162 (Area: MPKT_A_R)
prog_start.g_gruppe[25].nr 165 (Area: MPKT_KORR)
prog_start.g_gruppe[32].nr 140 (Area: CONT_MASK)
```

Group	Possible Default G-Functions
g_gruppe[0]	0, 1, 2, 3, 4 (G00, G01, G02, G03, G04)
g_gruppe[4]	17, 18, 19 (G17, G18, G19)
g_gruppe[6]	25, 26 (G25, G26)
g_gruppe[8]	51, 52 (G51, G52)
g_gruppe[12]	70, 71 (G70, G71)
g_gruppe[14]	90, 91 (G90, G91)
g_gruppe[21]	135, 137 (G135, G137)
g_gruppe[22]	138, 139 (G138, G139)
g_gruppe[23]	161, 162 (G161, G162)
g_gruppe[25]	164, 165 (G164, G165)
g_gruppe[32]	140, 141 (G140, G141)

- 默认进给速度

```
prog_start.feedrate      1000
```

当在执行 G01/G02/G03 代码时, 没有给定速度的话, 会默认初始速度为 1000mm/min。

- 设定速度曲线

`prog_start.slope.profile` 2

1) `profile`: 设定默认的加速度曲线 0: 阶跃响应; 1: 梯形曲线; 2: S 形曲线; 3: 梯形 HSC 曲线

2) 曲线类型设定可以在 NC 代码中单独设定 `#SET SLOPE PROFIL` 执行; 对于单独的轴曲线设定可以通过 `INDP_SYN` or `INDP_ASYN` 设定;

`prog_start.slope.ramp_time` 2

1) `ramp_time` 效力设定。0: 影响整个轴的加减速过程; 1: 影响轴的加速上升段时间; 2: 影响轴的加速下降段时间; 3: 减速上升段减速时间; 4: 减速下降段减速时间;

2) 与 `G132\G133\G134` 配合使用;

`prog_start.slope.acceleration` 2

1) 0: 影响轴的加减速度; 1: 影响轴的加速度; 2: 影响轴的减速度;

2) 与 `G130\G131` 配合使用;

- 进给速度单位

`prog_start.feedrate_factor` 0.1

进给速度单位 0.1 为 m/min, 100 为 mm/min, 默认为 100;

- 代码激活

`aep.g_gruppe[0-37]` 1

通过该参数激活相应的 G 代码, 只有激活参数之后才能够正常使用 G 代码;

G 代码表格如下:

Group	Contained G-Functions
<code>g_gruppe[0]</code>	G00, G01, G02, G03, G04, G33, G63, G74, G98, G99, G160, G301, G302
<code>g_gruppe[1]</code>	G08, G193
<code>g_gruppe[2]</code>	G09, G900, G901
<code>g_gruppe[3]</code>	G10, G11
<code>g_gruppe[4]</code>	G17, G18, G19
<code>g_gruppe[5]</code>	G20, G21, G22, G23, G351
<code>g_gruppe[6]</code>	G25, G26
<code>g_gruppe[7]</code>	G40, G41, G42
<code>g_gruppe[8]</code>	G51, G52
<code>g_gruppe[9]</code>	G53, ..., G59, G159
<code>g_gruppe[10]</code>	G60, G359, G360, G61, G260, G261
<code>g_gruppe[11]</code>	G166
<code>g_gruppe[12]</code>	G70, G71
<code>g_gruppe[13]</code>	G80 - G89, G800 - G819
<code>g_gruppe[14]</code>	G90, G91
<code>g_gruppe[15]</code>	G92
<code>g_gruppe[16]</code>	G93, G94, G95, G194
<code>g_gruppe[17]</code>	G96, G97, G196
<code>g_gruppe[18]</code>	G112

Group	Contained G-Functions
g_gruppe[19]	G115, G116, G117
g_gruppe[20]	G130, G131
g_gruppe[21]	G135, G136, G137
g_gruppe[22]	G05, G138, G139, G237, G238, G239
g_gruppe[23]	G161, G162
g_gruppe[24]	G163
g_gruppe[25]	G164, G165
g_gruppe[26]	G200, G201, G202
g_gruppe[27]	idle
g_gruppe[28]	G132, G133, G134
g_gruppe[29]	G150, G151
g_gruppe[30]	G100, G101, G102, G106, G107, G108
g_gruppe[31]	G12, G13
g_gruppe[32]	G140, G141
g_gruppe[33]	idle
g_gruppe[34]	G310
g_gruppe[35]	G167
g_gruppe[36]	not available
g_gruppe[37]	G129

- 定义宏命令

```
makro_def[6].symbol      CODE_2
makro_def[6].nc_code     G01 G91 X10 Y20 F100
```

通过宏命令可以定义参数的计算或者运动执行等；

- RTCP 空间坐标转换

```
kinematik_id 2 Default kinematic: 2
五轴系统选择坐标转换 ID 号 2 结构类型
#
kinematik[1].param[0] 1088000
kinematik[1].param[1] 1987000
kinematik[1].param[2] 342000
```

坐标转换 KINEMATIK ID 1 确定的各个相关点的数值，由结构类型决定。

- 显示子程序

```
suppress_prg_display_level      1
```

设置为 0 时，可以看到所有的子程序；1：仅显示主程序；1-50，可现实的子程序层级。

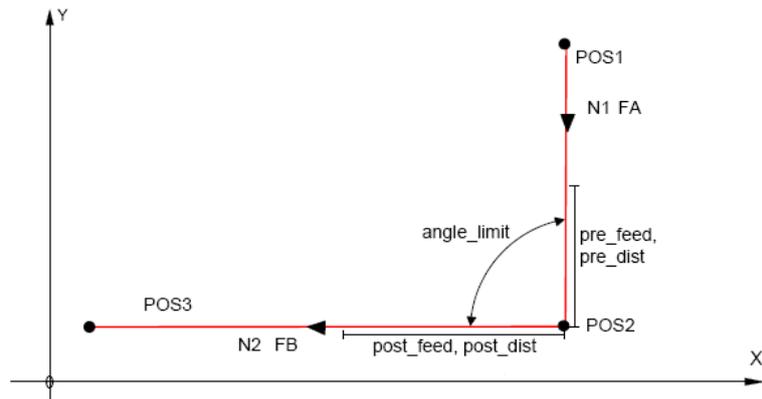
- 尖角控制功能

edge_machining.f_enable	0	[0: inactive 1: active]
edge_machining.angle_limit	1000000	[0—1800000]尖角角度设定
edge_machining.pre_dist	0	[0.1us]减速段距离
edge_machining.pre_feed	20000	[um/s]减速段速度
edge_machining.wait_time	1000000	[us]等待时间
edge_machining.post_dist	0	[0.1us]加速段距离
edge_machining.post_feed	1000	[um/s]加速段速度

1) 尖角控制功能是为客户后期开发的功能;

2) f_enable 是否使用尖角控制功能; angle_limit 角度限制; pre_dist 减速段距离; pre_feed 减速段速度; wait_time 等待时间; post_dist 加速段距离; post_feed 加速段速度;

3) 在 stupup list 添加 **【configuration.channel[1].path_preparation.function FCT_DEFAULT | FCT_EMF】**



- 小圆功能

#curve_dynamic_weighting_active	1	
#		
#curve_dynamic_weighting[0].radius_limit	30000	#[0.1um]radius limit
#curve_dynamic_weighting[0].velocity_fact	1000	#[0.1%]max speed*x%
#curve_dynamic_weighting[0].acceleration_fact	200	#[0.1%]max acc*x%
#curve_dynamic_weighting[0].ramp_time_fact	1000	#[0.1%]max ramp time*x%
#		
#curve_dynamic_weighting[1].radius_limit	70000	#[0.1um]
#curve_dynamic_weighting[1].velocity_fact	1000	#[0.1%]
#curve_dynamic_weighting[1].acceleration_fact	300	#[0.1%]
#curve_dynamic_weighting[1].ramp_time_fact	1000	#[0.1%]
#		
#curve_dynamic_weighting[2].radius_limit	100000	#[0.1um]
#curve_dynamic_weighting[2].velocity_fact	1000	#[0.1%]
#curve_dynamic_weighting[2].acceleration_fact	1000	#[0.1%]
#curve_dynamic_weighting[2].ramp_time_fact	1000	#[0.1%]

1) 小圆功能是为激光切割机床应用开发的一个功能

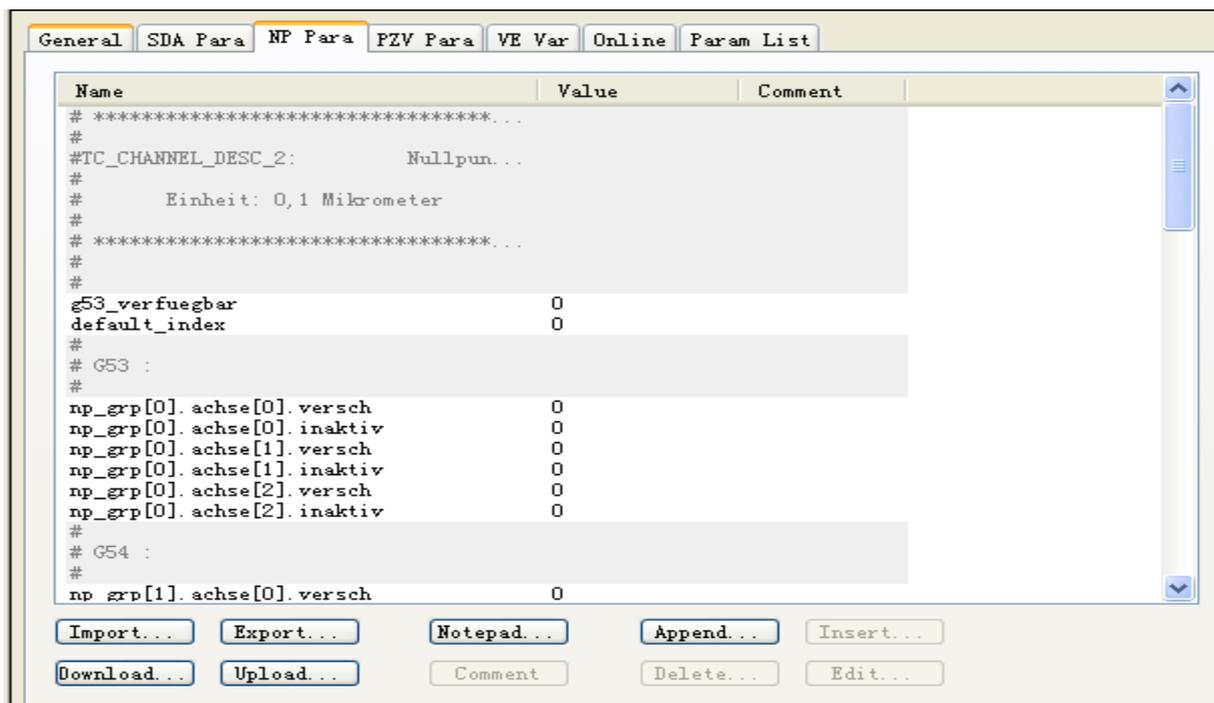
2) curve_dynamic_weighting_active 是否使用小圆功能;

3) radius_limit 小圆限制范围工三个范围 $0 < \text{radius limit} \leq 30000$, $30000 < \text{radius limit} \leq 70000$,

$70000 < \text{radius limit} \leq 100000$; velocity_fact 在相应的范围内快速进给速度的千分比; acceleration_fact 在相应的范围内快速进给加速的千分比; ramp_time_fact 在相应的圆范围内的快速进给加速时间的千分比。

Index	Radius limit value radius_limit [0.1 μm] (range limits)	Weighting factors velocity_fact [0.1%]	Weighting factors acceleration_fact [0.1%]	Weighting factors ramp_time_fact [0.1%]
0	10000 (0-10000)	100	10	3000
1	100000 (10000-100000)	500	100	2000
2	1000000 (100000-1000000)	1000	1000	1000

3、坐标系偏置



g53_verfuegbar 1
 default index 2

- g53_verfuegbar 设置的偏置值是否生效，0，不生效；1，生效；
- 设置的坐标偏置自动生效，自动生效哪个坐标偏置；

G53 :

np_grp[0].achse[0].versch 0 单位[0.1UM][0,0001°]
 np_grp[0].achse[0].inaktiv 0 0: 不生效; 1: 生效;
 np_grp[0].achse[1].versch 0 单位[0.1UM][0,0001°]
 np_grp[0].achse[1].inaktiv 0 0: 不生效; 1: 生效;
 np_grp[0].achse[2].versch 0 单位[0.1UM][0,0001°]
 np_grp[0].achse[2].inaktiv 0 0: 不生效; 1: 生效;

- np_grp[i], i 代表偏置组。如 np_grp[0]第 0 组偏置数据。
- achse[i], i 表示偏置轴号，如 achse[0]表示 X 轴偏置数据，achse[1]: Y; achse[2]: Z, 最大值为轴数；
- versch 表示偏置的距离或者角度，单位[0.1UM][0,0001°] ；
- inactive 表示相应轴的偏置是否生效。0: 不生效；1: 生效；
- 坐标偏置对应与不同的 G 代码 G53\G54\G55\G56\G57\G58\G59 以及 G159，其中 G159 与普通的坐标偏置不同之处在于，可以扩展坐标偏置值，如下图；

G function	Selection through G159	Zero offsets group index "I"
G53 or	G159 = 0	0
G54 or	G159 = 1	1
G55 or	G159 = 2	2
G56 or	G159 = 3	3
G57 or	G159 = 4	4
G58 or	G159 = 5	5
G59 or	G159 = 6	6
	G159 = 7	7
	G159 = 8	8
	G159 = 9	9
	G159 = 10	10

4、夹具位置偏置

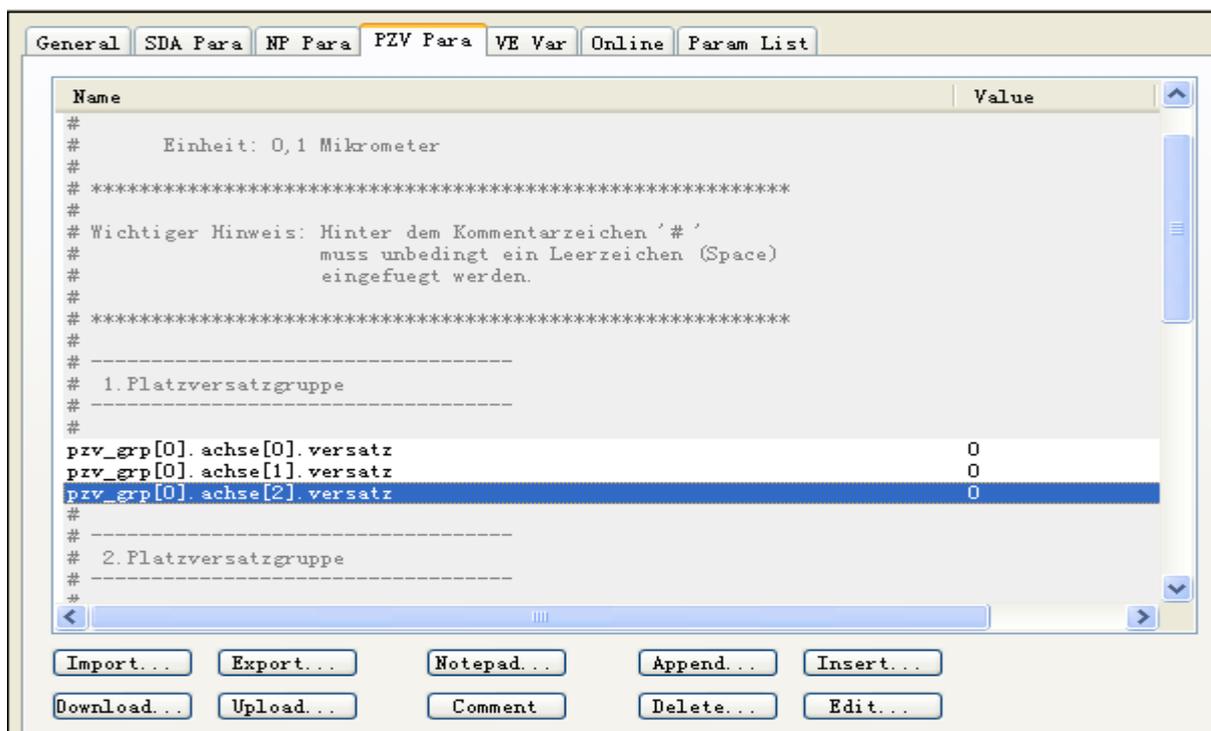
1.Group of clamp position offsets

pzv_grp[0].achse[0].versatz 100000 # Offset of 10 mm

pzv_grp[0].achse[1].versatz 200000 # Offset of 20 mm

pzv_grp[0].achse[2].versatz 300000 # Offset of 30 mm

- pzv_grp[i]: i 代表第几组偏置数据;
- achse[i]: i 代表第几根轴设置;
- versatz: 偏置数据单位 0.1um 或者 0.0001°
- 在 NC 代码中设置通过 PZV_INDEX 设置使用第几组偏置数据。



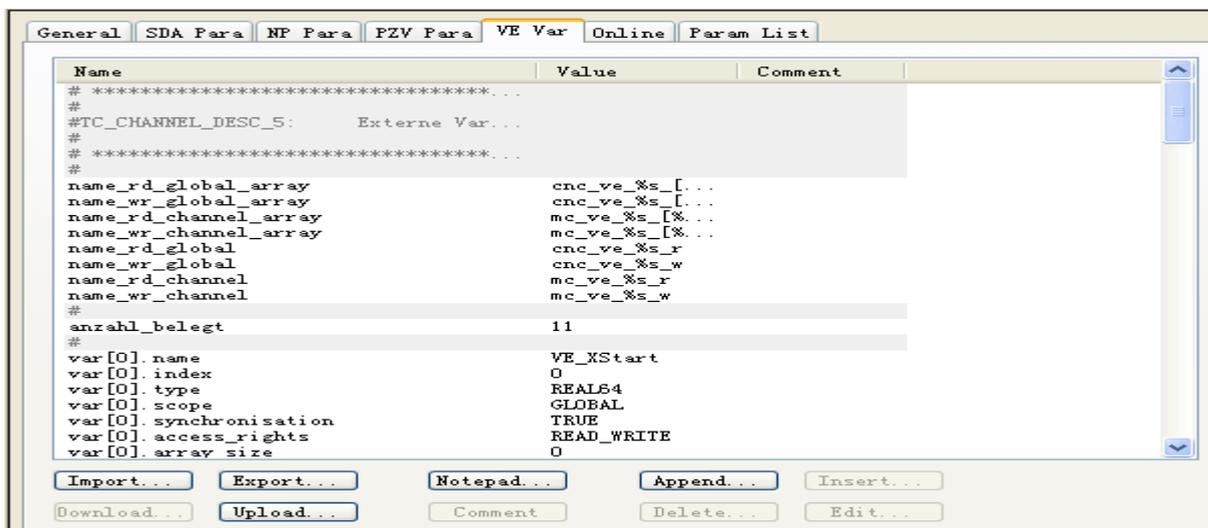
5、VE 变量

anzahl_belegt	11
● VE 变量最大数值, 要求设值必须大于等于最大的外部变量数;	
var[0].name	VE_XStart
var[0].index	0
var[0].type	REAL64
var[0].scope	GLOBAL
var[0].synchronisation	TRUE
var[0].access_rights	READ_WRITE
var[0].array_size	0
var[0].size	8
var[0].create_hmi_interface	1

- VE 变量的数量必须在 CNCTASK-GEO 的 HLI 中提前设定好, 设值时通过 stup VE 变量内存设定;
- var[i]: i 代表第几组 VE 变量序号;
- name: 外部变量名称, 用于 NC 代码中 V.E.name 设定;
- index: 定义当前内存区的外 i, 以 24byte 为单位;
- 设定变量 TYPE 类型 BOOLEAN、REAL64 等, size 为 1byte、8byte 等; 如下图

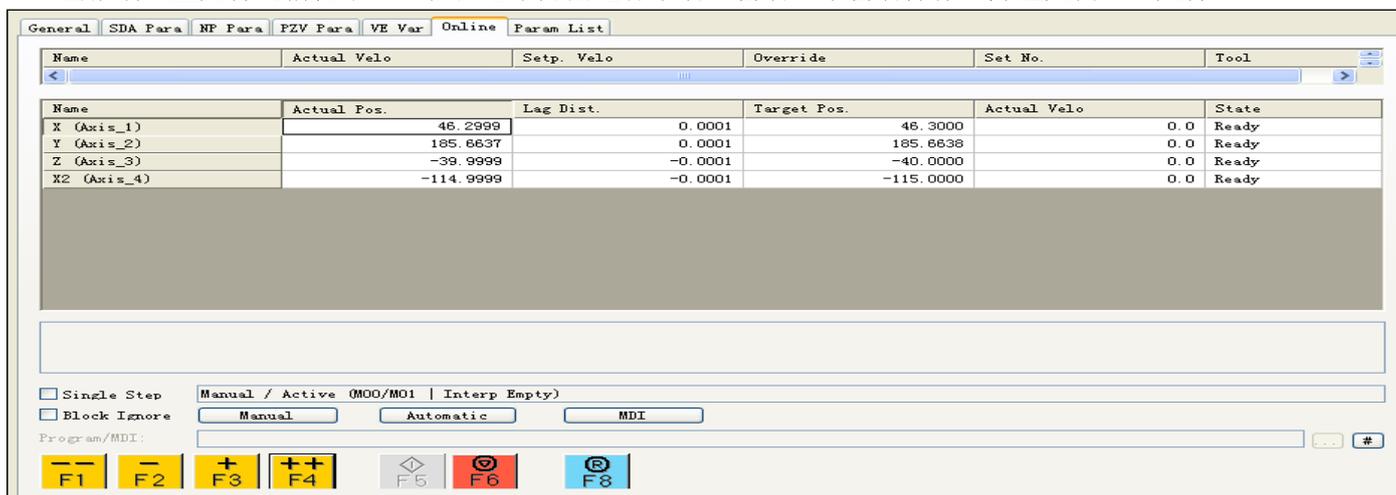
Variable type	Size
BOOLEAN, UNS08, SGN08	1 Byte
UNS16, SGN16	2 Byte
UNS32, SGN32	4 Byte
REAL64	8 Byte
STRING	128 Byte From CNC version V2.10.1025.00 onwards string variables can contain up to 128 characters including the end mark. For downgrade compatibility reasons for the use of string variables with more than 20 characters the parameter use_extended_string_var has to be set to 1. Depending on the parameter use_extended_string_var string variables are limited on the following length (including the end mark): use_extended_string_var = 0: 21 Byte (Default) use_extended_string_var = 1: 128 Byte

- Scope: 通道和全局查看;
- Synchronisation: 同步方式设定;
- Access_rights: 访问权限, 读写、只读、只写;
- Array_elements: 是否为数组结构, 如果为数组结构可定义数组大小, 反之赋值为 0;
- Size: 与 type 类型一致的 size 大小;
- Create_hmi_interface: 通过用户定义接口访问。

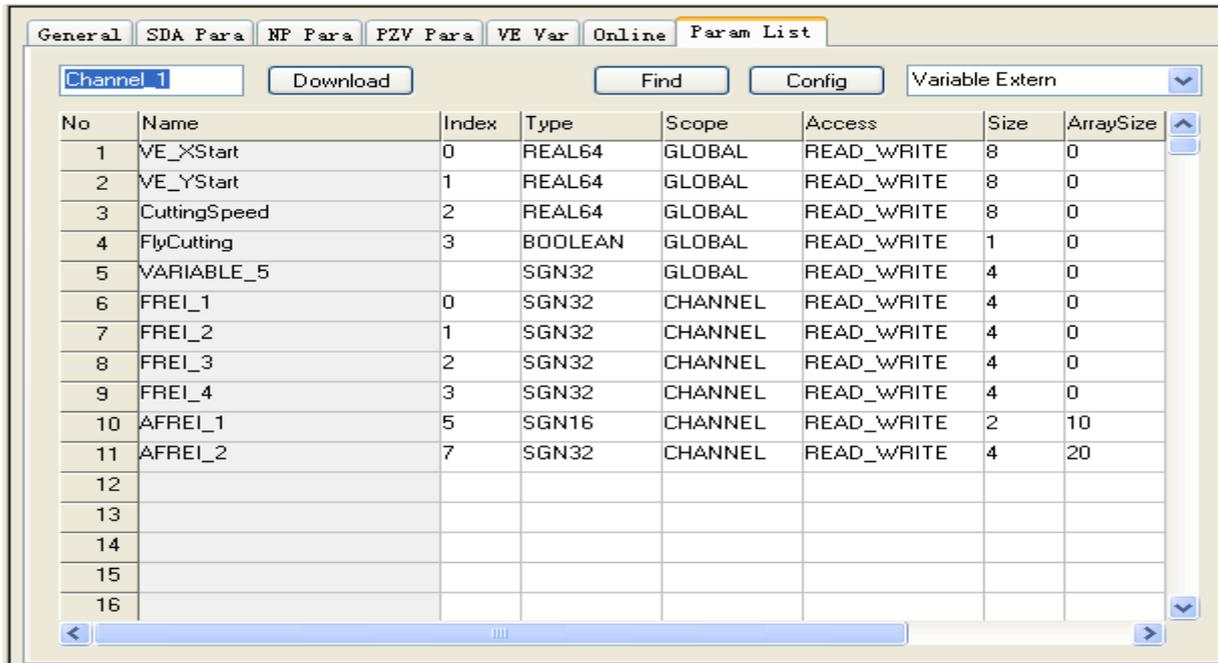


6、Online: 在线操作

当激活配置文件的情况下, 可以在这个界面进行手动、自动、半自动操作, 并且无需 PLC 文件。



7、Param List: VE 变量显示修改;



Channel 1 Download Find Config Variable Extern

No	Name	Index	Type	Scope	Access	Size	ArraySize
1	VE_XStart	0	REAL64	GLOBAL	READ_WRITE	8	0
2	VE_YStart	1	REAL64	GLOBAL	READ_WRITE	8	0
3	CuttingSpeed	2	REAL64	GLOBAL	READ_WRITE	8	0
4	FlyCutting	3	BOOLEAN	GLOBAL	READ_WRITE	1	0
5	VARIABLE_5		SGN32	GLOBAL	READ_WRITE	4	0
6	FREI_1	0	SGN32	CHANNEL	READ_WRITE	4	0
7	FREI_2	1	SGN32	CHANNEL	READ_WRITE	4	0
8	FREI_3	2	SGN32	CHANNEL	READ_WRITE	4	0
9	FREI_4	3	SGN32	CHANNEL	READ_WRITE	4	0
10	AFREI_1	5	SGN16	CHANNEL	READ_WRITE	2	10
11	AFREI_2	7	SGN32	CHANNEL	READ_WRITE	4	20
12							
13							
14							
15							
16							

五、轴参数

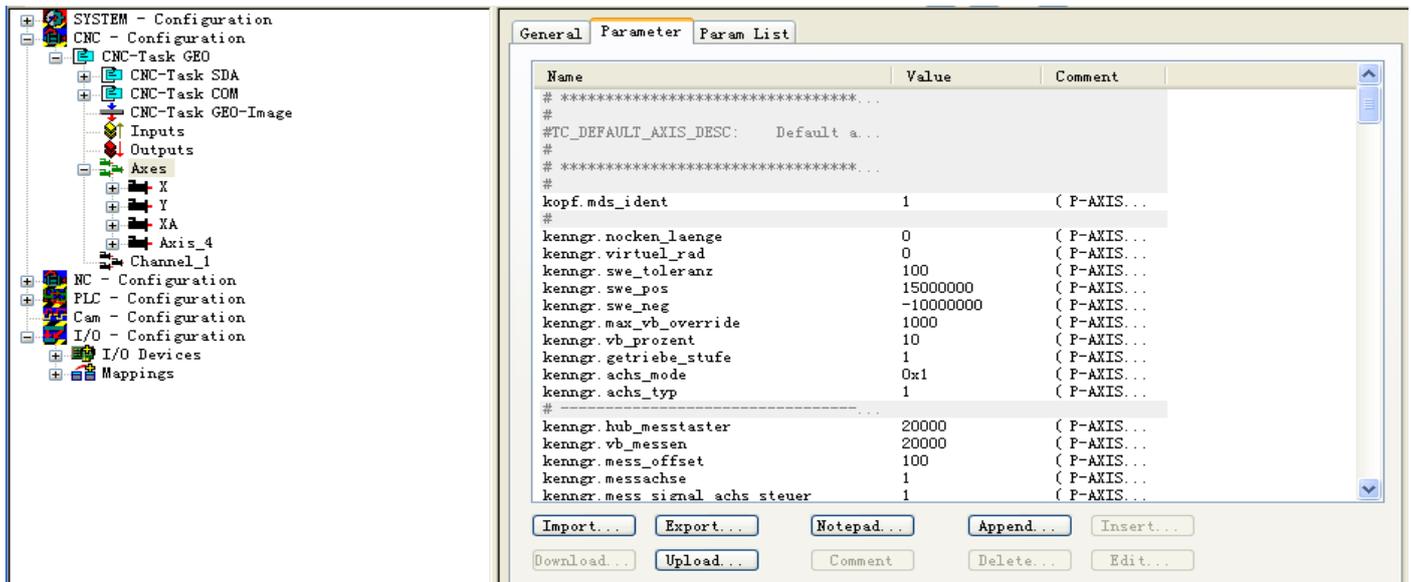
概述

在一个 CNC 工程中，需要包括一个通道以及一根 CNC 轴。如果通道提前加载的话，在添加轴的时候会默认将轴添加到当前通道中，如果先添加轴的话，需要单独在每个轴配置文件中单独设定。无论是否提前设定通道，都必须在添加轴的过程中，检查轴的通道属性。

在一个 CNC 工程中添加一根 CNC 轴的方法，在 Axes 处右击，添加 CNC 轴，轴参数为 Axes 中的默认参数，可在轴参数中对轴类型、动态特性、回参、脉冲当量等进行设定。

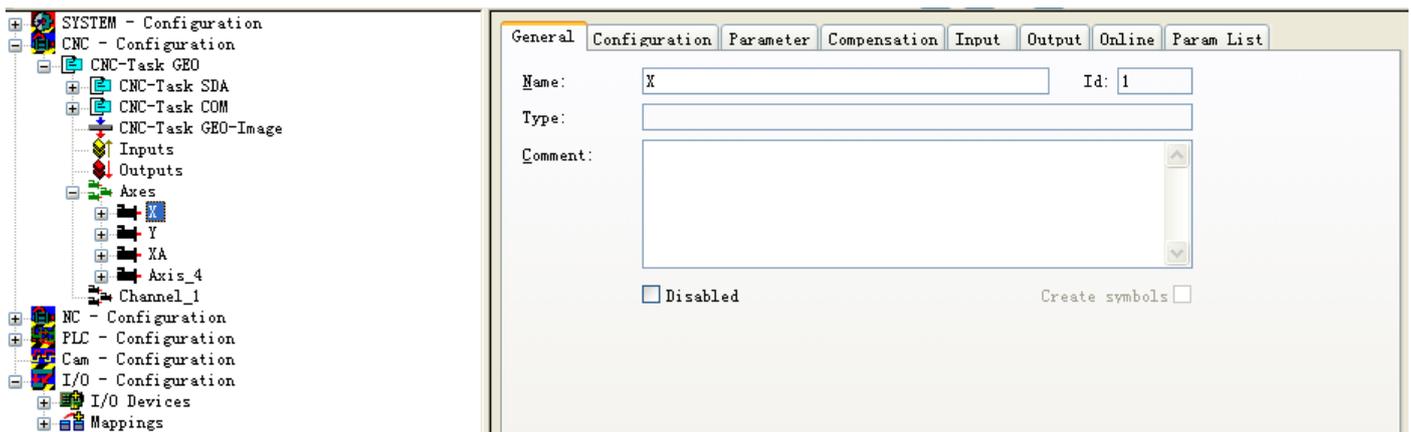
1、默认参数

在轴参数中，存在一个默认的轴参数，当添加轴的时候，会自动将轴参数调用到相应的轴中。默认参数与轴参数之间的区别在于，默认参数是对所有轴都生效的，与此同时，每根轴的轴参数会将默认的轴参数覆盖掉，也就是说最终生效的轴参数包括两部分，即轴参数设定部分，以及轴参数为设定部分而在默认轴参数中的设定的部分。另外一点就是 Param list 选项，这部分是将轴参数进行了表格化的处理。



2、添加一个 CNC 轴

右击  Axes 图标，添加一根 CNC 轴。



3、General

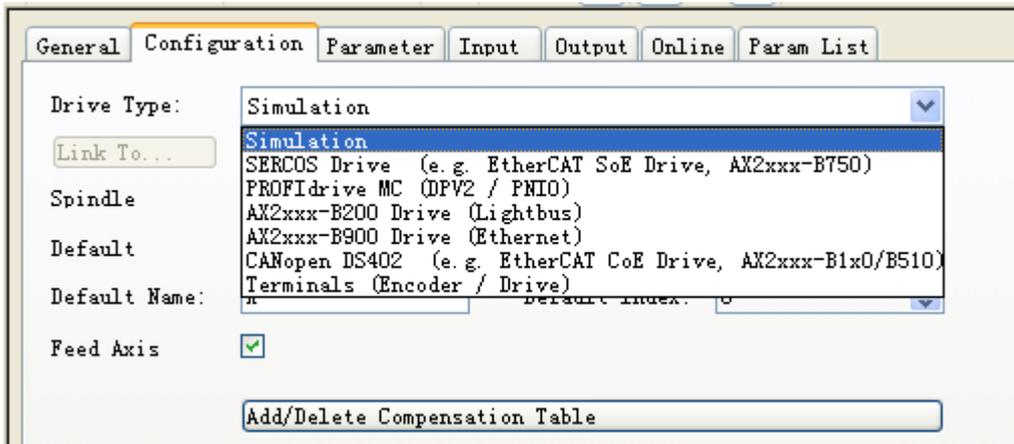
设定在配置文件中的显示名称，如 X\Y\Z 等，显示轴的 ID 号。

可以在此处配置，是否使用该 CNC 轴，即在 Disabled 是否选中判定该轴是否使用。

4、Configuration

- 硬件连接

设置轴属性，添加一个 CNC 轴，默认为虚拟轴，此轴没有外部 IO 链接，当然也可以添加实轴，选择轴类型如 SERCOS Driver、PROFIdrive、CANopen DS402、Terminals (Encoder/Drive) 等类型，通过 link to 连接实际的驱动器轴。需要注意的是要连接硬件的话，需要在配置文件中创建完成轴的配置，然后在 Link To 处连接实际的硬件。因为这些信息需要独到 CNC 配置中；



- SPINDLE

主轴设定，是否选择本轴为主轴，当轴设定为主轴的情况下，该轴不需要设定通道。

- 通道

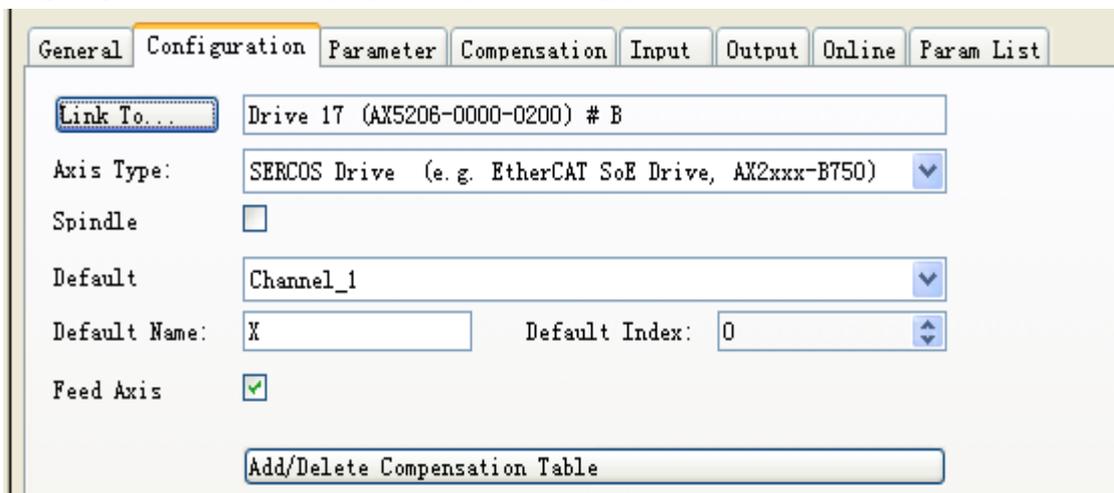
确定将轴分配到相应的通道中，这是一个默认的设置，在运行过程中可以通过 G 代码将轴分配到需要的通道中；

- 轴名称及轴号

在添加轴的时候，系统自定义一个轴名称依次 XYZ 等，可在此处修改，此处修改的轴名称意义在于 G 代码执行过程中，轴名称的标示作用。轴默认的 index 决定了轴的配置，实质上来讲，XYZ 轴的配置是通过这个 index 来完成，与之相对应的是轴名称。

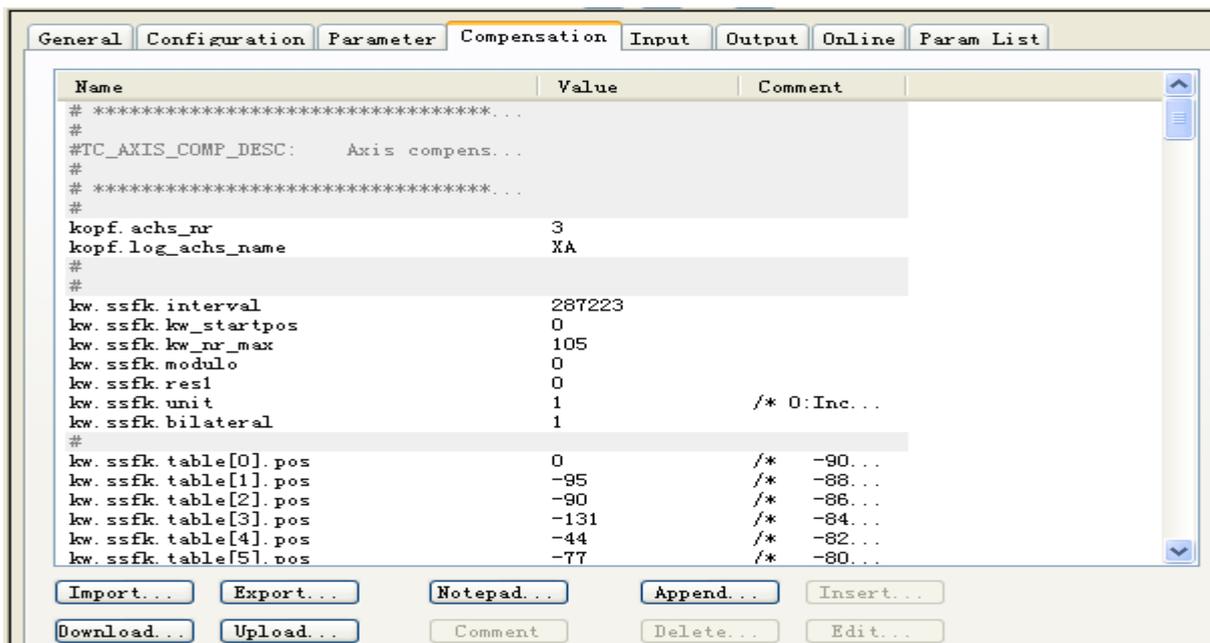
- 进给轴

设定轴是否是参与插补的，即使否参与轨迹方向上的速度分解。



- 螺距补偿表格

按下 **Add/Delete Compensation Table** 之后，添加一个螺距补偿表格；已有螺距补偿表格的情况下，删除该表格。



5、Parameter

CNC 轴参数主要设定轴的加减速、脉冲当量、最大速度、软限位、轴类型、龙门轴、跟随误差监控、各种补偿等；

- 自动生成参数

kopf.achs_nr	1	
kopf.mds_ident	1	
kopf.log_achs_name		X
antr.addroffs.input		0
antr.addroffs.output		0
antr.sercos.telegramm_typ		7
antr.sercos.ethercat		1
antr.sercos.at[0].ident_nr		51
antr.sercos.at[0].ident_len		4
antr.sercos.at[0].nc_ref		LAGEIST_WERT
antr.sercos.at[1].ident_nr		189
antr.sercos.at[1].ident_len		4
antr.sercos.at[1].nc_ref		S_0_0189
antr.sercos.mdt[0].ident_nr		47
antr.sercos.mdt[0].ident_len		4
antr.sercos.mdt[0].nc_ref		LAGESOLL_WERT
twincat.device.id		2
twincat.box.ads_port		1017
twincat.box.ads_netid		10.255.255.240.3.1
twincat.box.channel		1

#

这部分参数是自动生成的，主要工作在于读取轴配置中的设定，如轴号、轴名称以及对于硬件的链接读入。



- G00 动态设置
 - 最大速度
 - getriebe[0].vb_eilgang 1333333 # [um/s] G00 时速度
 - G00 轴加减速设定
 - getriebe[0].slope_profil.a_grenz 10000 # [mm/s2] G00 时加减速
 - getriebe[0].slope_profil.tr_grenz 100000 # [us] G00 时上升下降时间

- G01-G03 动态设定
 - 最大速度
 - getriebe[0].dynamik.vb_max 1333333 # [um/s] 最大速度
 - G01-G03 轴加减速设置
 - getriebe[0].slope_profil.a_beschl 12000 # [mm/s2]加速度
 - getriebe[0].slope_profil.a_brems 12000 # [mm/s2] 减速度
 - getriebe[0].slope_profil.tr_beschl_zu 100000 # [us] 加速过程上升时间
 - getriebe[0].slope_profil.tr_beschl_ab 100000 # [us] 加速过程下降时间
 - getriebe[0].slope_profil.tr_brems_zu 100000 # [us] 减速过程上升时间
 - getriebe[0].slope_profil.tr_brems_ab 100000 # [us] 减速过程下降时间

- 全局动态参数
 - G00-G01 加减速权重
 - #getriebe[0].dynamik.a_trans_weight 1000 # [us] 限制轮廓切换时加速度范围 0-1000;
此值为 0 时，关闭该参数。
 - #getriebe[0].dynamik.r_trans_weight 1000 # [us] 限制轮廓切换时的 Jerk 范围 0-1000,
 - getriebe[0].dynamik.tr_geom 100000 # [us] 轮廓转换时间
 - 加减速时间限制
 - getriebe[0].dynamik.tr_min 10000 # [us] 最小的上升时间
 - getriebe[0].dynamik.a_max 15000 # [mm/s2] 最大加速度

- 软限位
 - kenngr.swe_toleranz 5000 # [0.1um] 软限位误差容许值
 - kenngr.swe_pos 30050000 # [0.1um] 软件设定正限位
 - kenngr.swe_neg -50000 # [0.1um] 软件设定负限位

- 轴倍率
 - kenngr.max_vb_override 1000 # [0.1%]轴最大倍率主要用于独立轴以及主
轴

- 轴模式
 - kenngr.achs_mode 0x10001 # Achsmode 1=Linear, 4=Modulo
 - 本参数主要配置轴类型，包括直线轴、modulo 轴、主轴等
 - 0x00000001: 直线轴;
 - 0x00000004: modulo 轴，定长轴;
 - 以上两个类型必须选择其一，其他轴类型也要包含这个类型;
 - 0x00010000: 龙门轴主轴;
 - 0x00020000: 龙门轴从轴;
 - 龙门轴要定义是否是直线轴或者是 modulo 轴，如主轴 0x00010001，从轴 0x00020001;

- 轴类型
 - kenngr.achs_typ 1 # Achstyp 1=Linearachse, 2=Rundachse,
 - 轴类型三种: 0x0001: 直线轴; 0x0002: 旋转轴; 0x0004: 主轴;

- Modulo 轴，AX5000 伺服驱动配置参数

```

antr.mode_act_pos          1          # AX5000 伺服实际位置反馈
antr.mode_cmd_pos         1          # AX5000 伺服命令位置
这两个参数是对应于模式处理的，0：取决于轴类型；1：直线轴；2：旋转轴；
getriebe[0].moduloo       3600000   #(定长上限位 [0.0001°])
getriebe[0].modulou       0          #(定长下限位 [0.0001°])

```

设定 modulo 定长范围设置。

- 实际轴类型

```
kenngr.antr_typ          2
```

该参数自动或许当前轴连接的轴类型：

1：模拟量驱动轴轴搭配增量式编码器；2：SERCOS 电机类型；3：Profibus 驱动器；4：虚拟轴；5：Lightbus 伺服驱动；6：端子模拟量驱动搭配增量编码器；

- 龙门轴设置

从轴设置

```

kenngr.gantry_ax_nr      1          # 主轴轴号，
kenngr.gantry_max_diff_resetable 20000   # [0.1um] 可以消除的位置误差
kenngr.gantry_max_diff_reset_locked 30000   # [0.1um] 不可以消除的位置误差
kenngr.gantry_offset     0          # [0.1um] 主从轴位置固定误差
kenngr.gantry_vb_korr    0          # [um/s] 误差消除速度
kenngr.gantry_slave_no_homing 1       # 从轴不需要回参
kenngr.gantry_diff_check_without_homing 1   # 回参前就开始监控
kenngr.gantry_synchronous_slave_homing 0    # 如果是驱动器控制回参时才有意义
kenngr.cnc_controlled_stop_after_error 1    # 出错后 CNC 来控制停止

```

- 回参设置

回参功能设置

```

kenngr.ref_ri            0          # 回参时找参考点的轴运动方向；
kenngr.homing_without_zero_pulse 1       # 是否找 0 脉冲；
kenngr.fast_from_cam    1          # 回参过程中快速离开回参点；
kenngr.ref_ohne_nocken  0          # 回参过程中不使用参考点。
回参参考点，设置为 0；不使用参考点，设置为 1；
kenngr.vorz_richtung    0          #限制旋转轴的运动方向；
限制方向，设置为 1；不限制方向，设置为 0；
kenngr.beweg_ri         0          # 同上，旋转的方向限制。
配和上一参数，限制方向 1=正向限制, 0=负向限制；
kenngr.ref_ohne_rev     0          # 回参过程中不反向。
回参过程中方向设置为 0，回参过程中不方向为 1；
kenngr.homing_overflow_evaluation 0      # 回参结束后是否找编码器溢出，在增量式
编码器中，不能使用该值
antr.encoder_bit_range  20         # 编码器的位数，AX5000 统一为 20 位，该参
数与 homing_overflow_evaluation 为 1 和 homing_without_zero_pulse 为 0 时同时使用
kenngr.homing_type      CNC_CONTROLLED # 回参模式为 CNC 控制回参，AX5000 自己不
能回参
kenngr.fast_from_cam    0          #离开参考点时使用快速还是慢速 1=快速, 0 =
慢速
回参参数设置
getriebe[0].vb_refmax   100000     #回参快速[um/s]
getriebe[0].vb_reflow   2000       #回参慢速[um/s]
getriebe[0].pos_refpkt  0          # [0.1um]回参后位置

```

● 电机参数转换

getriebe[0].wegaufz	1048576	# [Inkr.] 一圈脉冲数 (分子)
getriebe[0].wegaufn	287231	# [0.1um] 实际运动距离, 也就是丝杠螺距 (分母)

电机脉冲当量设置, `wegaufz` 对应倍福伺服驱动器的编码分辨率, `wegaufn` 对应设置电机转一圈实际运动的距离, 注意单位为 0.1um。

getriebe[0].multi_gain_z	1800	# A\D 转换参数, AX5000 不用
getriebe[0].multi_gain_n	1	# A\D 转换参数, AX5000 不用

参数设置主要用于各类电机的应用, 以模拟量端子为例:

getriebe[0].multi_gain_z	32767	(DA-Digits)	/10V	1000 ° /min (标准值)
-----	=	-----	*	-----
getriebe[0].multi_gain_n	3000 (max speed -rev/min)		/10V	360° (电机一圈数值)

● 监控数值

到位监控

getriebe[0].window	100	# [0.1um] 时到位窗口
--------------------	-----	-----------------

跟随误差监控

getriebe[0].slep_ueberw_typ	4	# 类型 4 为速度无关的 Lag 监控,
getriebe[0].slep_min	5000	# 静止时最大允许 Lag 0.1um
getriebe[0].slep_max	10000	# 运动时最大允许 Lag
antr_digital.nbr_delay_cycles	3	# AX2000 为 4, AX5000 为 3
lr_param.suppress_pos_lag_error	0	# 0 为有位置监控, 1 为关闭 Lag 监控

● 电机换向

lr_hw[0].vz_stellgr	0	# 设定值反向
lr_hw[0].vz_istw	0	# 实际值反向

注意: CNC 中的电机换向类似于 NC 中的电机换向编码器、电机方向必须同时换向, 否则会发生飞车, 注意安全。

● 手动

handbetrieb.hb.vb_max	1333333	# [um/s] (G200) 手动最大速度
handbetrieb.hb.a_max	2000	# [mm/s ²] (G200) 手动最大加速度
handbetrieb.hb.tr	260000	# [us] 手动上升时间

手动设置, 手动设置数值必须小于最大速度、加速度、加速时间等。

● 螺距补偿

lr_param.ssfk	1
---------------	---

螺距补偿表格是否生效, 0: 螺距补偿表格不生效; 1: 螺距补偿表格生效。

6、Compensation 螺距补偿表格

- 补偿轴

kopf.achs_nr	3	
kopf.log_achs_name	XA	

 自动识别补偿轴名称以及轴号；
- 补偿轴参数

kw.ssfk.interval	287223	
------------------	--------	--

 补偿间隔设置，通常设置为轴转一圈的距离即脉冲当量；

kw.ssfk.kw_startpos	0	
---------------------	---	--

 补偿起点设置，设置轴补偿起点；

kw.ssfk.kw_nr_max	105	
-------------------	-----	--

 补偿点数设定，即补偿点数；

kw.ssfk.modulo	0	
----------------	---	--

 补偿轴类型，是否为 modulo 轴；

kw.ssfk.unit	1	/* 0:Increments 1:Metric */
--------------	---	-----------------------------

 补偿单位，0: 脉冲；1: 米制

kw.ssfk.bilateral	1	
-------------------	---	--

 补偿方式，0: 单向补偿；1: 往复补偿；

kw.ssfk.table[0-104].pos	0	
--------------------------	---	--

 正向补偿值设定，单位 0.1um；

kw.ssfk.table[0-104].neg	0	
--------------------------	---	--

 负向补偿值设定，单位 0.1um；

7、input

CNC 轴输入值；

Name	Type	Id	Ref
Drive status word	UINT 16	S-0-0135	<hidden>
WcState	UINT 16	P-7-4093	P_7_4093
Position feedback 1 value	INT 32	S-0-0051	LAGEIST_WERT
Following distance	INT 32	S-0-0189	S_0_0189

8、output

CNC 轴输出值

Name	Type	Id	Ref
Master control word	UINT 16	S-0-0134	<hidden>
Reserved	UINT 16	S-0-0000	S_0_0000
Position command value	INT 32	S-0-0047	LAGESOLL_WERT

9、Online 在线监控

在线将空当前轴位置以及设定位置；

10、Param List

轴参数表格化设置。

No	Display Name	Value	Unit	Comment
1	Starting up			
2	Operating mode of the axis	0x20001	-	Attributes of axis
3	Axis type	1	-	Axis type of axis (Linear, Rotatory, spindle)
4	Absolute path measurement system		-	
5	Proportional factor K _v		1/s	Position control loop, not SERCOS
6	Path resolution of the measuring system numerator	1048576	Inkr.	Part of scaling factor
7	Path resolution of the measuring system denominator	28.72310	mm	Part of scaling factor
8	D/A-adaption factor numerator	1800	Bit	Position control, no Sercos, no CAN-open
9	D/A-adaption factor denominator	1	-	Position control, no Sercos, no CAN-open
10	Upper modulo limit		0.0001°	Rotary axis
11	Lower modulo limit		0.0001°	Rotary axis
12	Fading out the position lag error	0	-	
13	Compensation value of D/A converter drift		Bit	
14	Sign reversal of command value	0	-	
15	Sign reversal of actual value	0	-	
16	Restriction of direction of rotation	0	-	Rotary/spindle axis, only one turning direction
17	Preferred direction of rotation	0	-	Rotary/spindle axis

本章总结：

本章以配置一个 CNC 轴为主线，从添加一个 CNC 任务，到添加一个 CNC 通道，再到添加 CNC 轴的顺序来讲解了每部分对应的参数设置信息，同时这也是我们在新建一个 CNC 工程时所遵循的步骤。

我们从中可以看到 CNC 任务着重对于 CNC 运行周期以及初始化配置文件、路径规划、通讯设置等；通道设置，主要的工作在于对于 M 函数、子程序、坐标偏转、坐标系转换、VE 变量等具体程序的系统参数设置；轴参数设置主要是对运动轴本身进行设置，包括动态参数、脉冲当量、软限位、监控参数设置以及补偿设置。

第五章 TwinCAT CNC 工程 PLC 基础程序

概述:

本章旨在介绍如何配置如何新建一个 CNC 工程 PLC 程序文件，并对 PLC 程序进行讲解。

PLC 程序包括两部分，一部分是倍福提供的标准程序接口处理程序，另一部分是客户自己编辑的程序。

PLC 程序介绍:

倍福 CNC 工程中的 PLC 程序部分，提供一个 CNC 标准框架，这个框架是对于倍福 CNC 接口的处理。包括初始化、操作模式部分、控制功能部分、MHT 函数部分、VE 变量部分、功能块查询、报警程序、轴设置、通道设置、手动设置等。下面将以 main 主程序为主线，针对程序进行讲解。

一、MAIN 主程序

通常主程序以 MAIN 程序为开始，设定 TASK 周期优先级，在 CNC 周期设置方面，通常设为 1-2ms。

具体程序如下：

MAIN:

```

PRG_ShowMessages();           (*报警显示程序*)
HLI_Init();                   (*初始化程序部分*)
IF CNCSystem.bHLIInit THEN    (*程序初始化完成后才可以调用*)
    PRG_AxesHandler();        (*轴操作设置*)
    PRG_ManualMode();        (*手动操作*)
    PRG_ChannelHandler ();    (*通道功能写入*)
END_IF

```

1) 报警信息显示程序

报警显示程序设定报警组别、类型、优先级等设定；

2) 初始化程序部分

初始化程序部分，是通过读写 HLI 接口，对 CNC 进行初始化的过程，对轴、通道、VE 变量等进行读写。

3) 初始化完成后执行调用

建议所有的程序都放在初始化完成之后执行，尤其是涉及到轴或者通道执行的部分；

4) 轴操作相关

基础程序轴操作部分包括对所有 CNC 轴进行使能，回参等操作；

5) 手动操作

基础程序手动操作部分主要对于 CNC 轴进行手动操作，包括选择寸动、联动、手轮操作等模式。部分功能轴组、通道选择、快慢速设定、速度设定等；

6) 通道功能写入

定义通道功能，并连接相应变量。通道倍率设定，可以实现通道超调如 120%等。设定当前模式手动、自动、半自动、回参、开始、暂停、复位包括调用程序的地址、回退、NC 代码块查询等功能。

二、报警信息显示

报警信息显示包括三部分程序，第一部分：报警设定；第二部分：报警部分；第三部分：报警显示界面；

第一部分代码：

报警类型：

TCEVENTCLASS_NONE	:=0,	No class (无关提示)
TCEVENTCLASS_MAINTENANCE	:=1,	Maintenance hint (维护提示)
TCEVENTCLASS_MESSAGE	:=2,	Message (信息)
TCEVENTCLASS_HINT	:=3,	Hint (提示)
TCEVENTCLASS_STATEINFO	:=4,	State information (状态信息)
TCEVENTCLASS_INSTRUCTION	:=5,	Instruction (指示)
TCEVENTCLASS_WARNING	:=6,	Warning (警告)
TCEVENTCLASS_ALARM	:=7,	Alarm (故障)
TCEVENTCLASS_PARAMERROR	:=8	Parameter error (参数错误)

具体程序：

```

IF NOT bStart THEN
  bStart := TRUE;
  SimpleEventLog1 (Init           := TRUE,
    SourceId           := 51,
    Class              := TCEVENTCLASS_WARNING,
    QuitRequiredD     := TRUE,
    adrAlarmArray     := ADR(MSGList1),
    adrQuitArray      := ADR(MSGListQuit1)
  );
  SimpleEventLog2 (Init           := TRUE,
    SourceId           := 50,
    Class              := TCEVENTCLASS_ALARM,
    QuitRequired      := FALSE,
    adrAlarmArray     := ADR(MSGList2),
    adrQuitArray      := ADR(MSGListQuit2)
  );
  SimpleEventLog3 (Init           := TRUE,
    SourceId           := 13,
    Class              := TCEVENTCLASS_INSTRUCTION,
    QuitRequired      := FALSE,
    adrAlarmArray     := ADR(MSGList3),
    adrQuitArray      := ADR(MSGListQuit3)
  );
ELSE
  SimpleEventLog1 (Init           := FALSE);
  SimpleEventLog2 (Init           := FALSE);
  SimpleEventLog3 (Init           := FALSE);
END_IF

```

- 1) SourceId, 报警程序以 SourceID 号码分组, 可以任意定义报警组;
- 2) Class, 报警类型选择, 可设置 9 种报警类型;
- 3) QuitRequired, 报警是否确认, 报警不因报警信号消除而消失, 必须经过确认才能取消;
- 4) adrAlarmArray, 报警数组, 根据报警数组的组号确定报警 ID;
- 5) adrQuitArray, 报警确认数组, 与报警确认相关联;

第二部分:

报警程序编辑要根据外部信号对报警组号 adrAlarmArray 完成报警 ID 的输出;

```
如: IF bXLimit THEN
      MSGList2 [11]:=TRUE;
    ELSE
      MSGList2 [11]:=FALSE;
    END_IF
```

本程序完成了一个报警级别为 TCEVENTCLASS_ALARM, 报警 ID 序号为 11, 不需要确认的报警信息输出。

第三部分:

报警信息输出到 HMIPRO 中, 设定方法键第二章, 第四部分的报警信息编辑操作即可。

❗ X轴正向限位报警						CNC	
通道 1	P I轴	实际位置	滞后距离	剩余距离	Override		
	X	mm	0.000	0.000	0%	130%	单步运行
	Y	°	0.000	0.000	0%	120%	忽略行
	Z	mm	0.000	0.000	0%	100%	

三、初始化程序

初始化程序必须在执行与 CNC 相关程序之前执行，且 PLC 执行时，只执行一次。完成对于 HLI 的读写以及对刀具功能、轴功能、通道功能的初始化设定。

```

HLI_Init:    (执行 CNC 的初始化程序)
    HLI_InitToolConfig();                (*初始化刀具信息*)
    HLI_SetToolData();
    Hli(bStart:=SystemTaskInfoArr[1].firstCycle);    (* PLC 第一次周期运行 *)
    IF Hli.bInitialized = TRUE AND Hli.bError = FALSE THEN (* 初始化成功 *)
        IF CNCSystem.bUserInitDone = FALSE THEN
            CNCSystem.bUserInitDone := HLI_UserInitialisation();
            ADSLOG(strMsg:=' High Level Interfaces initialized: %s', nDINT:=0, strArg:=sVersion);
        END_IF
        CNCSystem.bHLIInit := TRUE;
    ELSE                                        (* 初始化失败报警*)
        MessageOutput(CLK:=Hli.bError);
        IF Hli.bError = TRUE THEN
            IF MessageOutput.Q THEN
                ADSLOGDINT( msgCtrlMask := ADSLOG_MSGTYPE_ERROR OR
                    ADSLOG_MSGTYPE_MSGBOX OR ADSLOG_MSGTYPE_LOG,
                    msgFmtStr := 'Fehler beim Initialisieren des HLI-Interfaces Fehlercode: %d',
                    dintArg := Hli.iErrorId);
            END_IF
            CNCSystem.nInitErrorId := Hli.iErrorId;
            CNCSystem.bInitError := TRUE;
        END_IF;
    END_IF;

```

涉及程序框架部分包括如下子程序

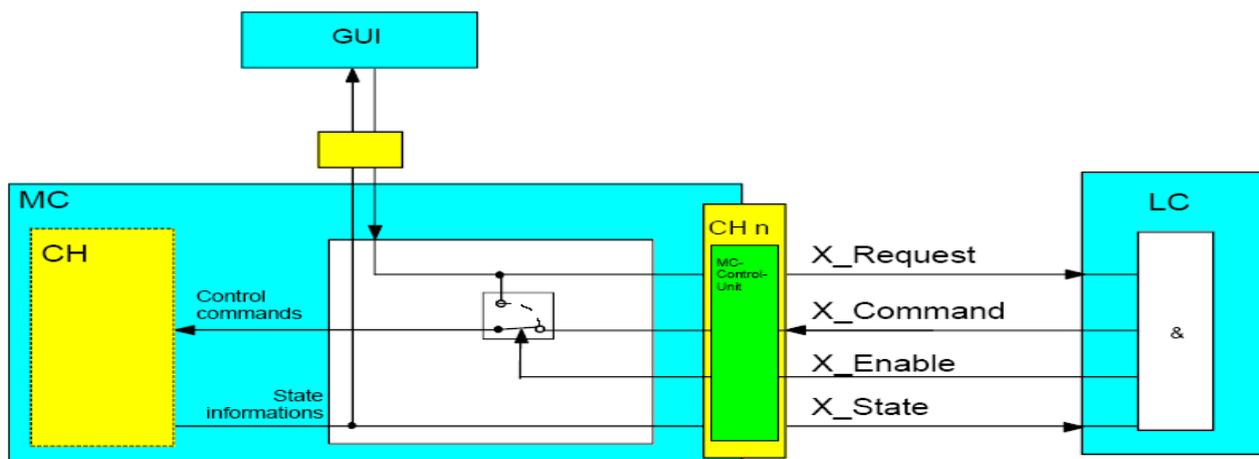
```

功能块: CNC→1 Init→ HLI_Init
        → HLI_InitToolConfig
        → HLI_SetToolData
        → HLI_UserInitialisation
        → HLI_SetAxisControlToPlc
        → HLI_SetChannelControlToPlc

```

CNC 功能开启介绍:

轴设置与通道设置中都需要对 HLI 进行操作与 CNC 进行命令交互，譬如轴设置中的轴使能命令、通道中的倍率命令、复位命令等。无论是人机界面还是 PLC 程序开启相应的 CNC 功能，都需要通过设置相应命令的使能信号即 **X_Enable** 为 TRUE, 然后才能够写 **X_Command** 到 CNC 中，执行相应功能。其流程图如下:



四、轴使能设置

轴使能设置主要对于轴使能，可在其中将回参等动作放在一起。

程序:

```

IF NOT bFirstCycle THEN                                (*第一个周期，自动启动使能*)
    bFirstCycle:=TRUE;
    PLCAxisEnable:=TRUE;
END_IF
fbTonDriveOn(IN:= PLCAxisEnable, PT:= T#1s);
IF PLCAxisEnable THEN
    bTorquePermission      := TRUE;
    bFeedhold              := TRUE;
    IF fbTonDriveOn.Q THEN
        bDriveOn           := TRUE;
    END_IF
ELSE
    bTorquePermission      := FALSE;
    bFeedhold              := FALSE;
    bDriveOn               := FALSE;
END_IF
FOR nLogAxisNo := 1 TO UDINT_TO_INT(nAxis) DO
    pArrAxis[nLogAxisNo] :=pAC[nLogAxisNo];
    HLI_SetAxisTrackingOperation(Axis:= nLogAxisNo, bPLCAxisEnable:= PLCAxisEnable);
    HLI_SetAxisFeedHold (Axis:= nLogAxisNo, TorquePermission:= bTorquePermission, Feedhold:=
bFeedhold, DriveOn:= bDriveOn);
END_FOR

```

程序注释

1) 使能控制设置是否为开环控制，对应到子程序对 HLI 进行设置。调用 CNC 程序框架中的子程序。

HLI_SetAxisTrackingOperation:

```
(* tracking operation (follow up) on / off *)
IF NOT bPLCAxisEnable THEN
    pAc[Axis]^ .addr^.McControlLr_Data.MCControlBoolUnit_FollowUp.X_Command := TRUE;
    HLI_SetAxisTrackingOperation := TRUE;
ELSE
    pAc[Axis]^ .addr^.McControlLr_Data.MCControlBoolUnit_FollowUp.X_Command := FALSE;
    HLI_SetAxisTrackingOperation := FALSE;
END_IF
```

2) 使能设置，调用子程序 HLI_SetAxisFeedHold，控制 CNC 命令实现使能，包括扭力、使能开等；

HLI_SetAxisFeedHold:

```
(* set regulation and feed *)
pAc[Axis]^ .addr^.McControlLr_Data.MCControlBoolUnit_TorquePermission.X_Command :=
TorquePermission;
pAc[Axis]^ .addr^.McControlLr_Data.MCControlBoolUnit_ReleaseFeedhold.X_Command := Feedhold;
pAc[Axis]^ .addr^.McControlLr_Data.MCControlBoolUnit_DriveOn.X_Command := DriveOn;
```

涉及程序框架部分包括如下子程序

功能块: CNC→1 Init→HLI_SetAxisControlToPlc

该部分定义启动功能

功能块: CNC→3 Control→HLI_SetAxisFeedHold

功能块: CNC→3 Control→HLI_SetAxisTrackingOperation

这两部分分别对与控制方式以及扭力、使能进行设定。

五、手动操作

手动模式下，以轴序号为标志通过 HLI 设置轴运动类型、快慢速速度设定、手动方向、速度等；

手动模式类型：

- 1 - Handwheel 手摇轮，移动位置及速度取决于手摇轮；
- 2 - Tip mode 连续运动，以参数设定速度，按键操作决定移动长度；
- 3 - Jog mode 寸动模式，以参数设定速度，预先设置长度，按键操作；

初始设定为手动模式：

```
IF NOT INIT THEN
    PLCMachineMode[1].Manual := TRUE;
    INIT := TRUE;
END_IF
(*选择手动模式类型*)
IF PLCManualTyp_Handwheel THEN
    stManualModeParameter[1].stChannelParameter.nManualTyp := 1;
ELSIF PLCManualTyp_Tip THEN
    stManualModeParameter[1].stChannelParameter.nManualTyp := 2;
ELSIF PLCManualTyp_Jog THEN
    stManualModeParameter[1].stChannelParameter.nManualTyp := 3;
ELSE
    stManualModeParameter[1].stChannelParameter.nManualTyp := 2;
END_IF
```

选择手动轴：

```
stManualModeParameter[1].stChannelParameter.nChan           := 1; (*通道 1*)
stManualModeParameter[1].stChannelParameter.nControlUnits   := 1; (*调用 MANUALMDS 表格*)
```

快慢速设定：

```
IF PLCManualRapidMode THEN
    Speedstate:=1;
ELSE
    Speedstate:=0;
END_IF
```

手动速度：

```
1XAxisManualSpeed[0]:=PLCParList1[1]*50000/3;
1XAxisManualSpeed[1]:=PLCParList1[2]*50000/3;
```

寸动距离设定:

```
stManualModeParameter[1].stChannelParameter.nInc:=PLCManual_JogIncrement*10; (**0.1 um **)
```

手动操作:

- ◆ 选择执行轴:
stManualModeParameter[1].stAxisParameter.stControlUnit[1].nLogAxisNumber :=
GD_VisuAxisNumber;
- ◆ 手摇轮设置: 分辨率+手轮位置
stManualModeParameter[1].stChannelParameter.nHandwheelResolution:=iHandwheel_Rate;(0.1 um)
stManualModeControlElements[1].nHandwheelInc[1]:=LREAL_TO_DINT(HandWheel.NcToPlc.ActPos);
- ◆ 手轮操作: =手动速度+正负方向
stManualModeParameter[1].stAxisParameter.stControlUnit[1].nSpeed
:=LREAL_TO_DINT(1XAxisManualSpeed[Speedstate]);
stManualModeControlElements[1].stControlKeys[1].bKeyPos:=PLCManualAxisPlus; (*X +*)
stManualModeControlElements[1].stControlKeys[1].bKeyNeg:= PLCManualAxisMinus; (*X-*)

涉及程序框架部分包括如下子程序

功能块: CNC→1 Init→HLI_SetChannelControlToPlc

该部分是提前打开相应功能接口如手轮等

功能块: CNC→2 OperationMode→ HLI_CNCChannel

该部分是设定手动模式;

功能块: CNC→2 OperationMode→ HLI_ChannelModeManual

该部分是各个模式下的手动参数选择;

功能块: CNC→2 OperationMode→ModeManual→ HLI_ChannelModeManual2

该部分是手动功能操作;

功能块: CNC→2 OperationMode→ModeManual → HLI_AxisInManualMode

该部分手动轴选择;

功能块: CNC→2 OperationMode→ModeManual → HLI_ManualActivation

该部分手动模式选择以及参数如: 手轮倍率、手动速度等设定

功能块: CNC→2 OperationMode→ModeManual → HLI_ManualDeactivation

该部分手动模式关闭;

功能块: CNC→2 OperationMode→ModeManual → HLI_HLI_ManualExecution

该部分手动执行

功能块: CNC→2 OperationMode→ModeManual → HLI_ManualExecutionRapidMode

该部分手动快速执行

功能块: CNC→2 OperationMode→ModeManual → HLI_SetAxisOverride

该部分设定手动运行的倍率

功能块: CNC→2 OperationMode→ModeManual → PRG_AnalogJoystick

该部分模拟量驱动

程序: SystemProgram→ Basic→ PRG_ManualModeHMI

该部分设定手动运行参数;

程序: SystemProgram→ Basic→ PRG_ChannelHandler

该部分将手动参数发送到手动程序进行执行;

涉及到的轴命令、状态:

```
pAC      : ARRAY[1..HLI_SYS_AXMAX] OF POINTER TO HLI_AXIS_ADDRESS;
```

```
HLI_AXIS_ADDRESS:
```

```
  addr    : POINTER TO AxeData;
```

```
AxeData:
```

```
  StateLR_Data    : StateLR;
```

```
pAC[1]^ . addr^ . StateLR_Data.
```

```
StateLR:
```

```
LogName      : TYP_10_HLIAxeName; (*logical axis name, see axis list*)
X_RevControlActiv  : HLI_BOOLEAN; (*Drehzahlueberwachung aktiv DZ_UEBERW_AKTIV 0x00000001*)
X_RevReached      : HLI_BOOLEAN; (*Soll Drehzahl erreicht DZ_ERREICHT 0x00000002*)
X_RevZero         : HLI_BOOLEAN; (*Drehzahl ist Null DZ_NULL 0x00000004*)
X_RevControlInvalid : HLI_BOOLEAN; (*Drehzahlueberwachung ungueltig, da Regelgrenze ueberschritten wurde DZ_UEBERW_??? 0x00000008*)
D_CurrentRev      : HLI_SGN32; (*--- Member cmd_rev ---ehem. vb_sollwert enthaelt bei SPDL Soll Drehzahl*)
D_ActiveRev       : HLI_SGN32; (*--- Member act_rev ---ehem. vb_istwert enthaelt bei SPDL Istdrehzahl*)
D_CmdRev          : HLI_SGN32; (*programmed revolution M03, M04*)
D_FollowingError  : HLI_SGN32; (*Schleppfehler*)
D_BlockNr         : HLI_SGN32; (*Satz-Nummer*)

D_CurrentPositionACS : HLI_SGN32; (* Istwert der Achse, intern: m_istw_absolut*)
D_ActivePositionACS  : HLI_SGN32; (* Sollwert der Achse, intern: m_sollw_absolut*)
D_ActivePositionACS0 : HLI_SGN32; (* Sollwert der Achse vor der Achstransformation, intern: m_sollw_absolut_acs0*)
D_EndPositionACS     : HLI_SGN32; (*Zielposition der Interpolation*)
D_PositionOffsetPhysicalLogicalACS : HLI_SGN32; (*Positionoffset zwischen physikalischer und logischer Interpolator Achse *)

W_NCChannel       : HLI_UNSI6; (*Kanalnummer zu der die Achse momentan gehoert.*)
W_AxisState       : HLI_UNSI6; (*Zustand der Achse. Aus enum HLI_AXIS_STATE*)
W_DriveType       : HLI_UNSI6; (*"kenngr. antr_typ" aus achsmds.lis (Definitionen siehe spez_mds.inc)*)
W_LinkToLogicalAxis : HLI_UNSI6; (* link of real axis to logical interpolator axis, == 0 -> not linked *)

X_HomingDone      : HLI_BOOLEAN; (*---*)
X_InWindow        : HLI_BOOLEAN; (*Achse im Regelfenster*)
X_InPosition      : HLI_BOOLEAN; (*Achse in Position*)
X_IsMoved         : HLI_BOOLEAN; (*My? Achse ist bewegt (satzbezogen)*)
X_IsMoving        : HLI_BOOLEAN; (*My? Achse wird verfahren*)
X_IsMovingForward : HLI_BOOLEAN; (*Achse wird in Vorwaertsrichtung verfahren*)
X_FillUp1         : HLI_BOOLEAN; (*fill-up members FOR MOD 4 alignment*)
X_FillUp2         : HLI_BOOLEAN; (*fill-up members FOR MOD 4 alignment*)
D_Word1           : HLI_UNSI32; (*Data transmitted via the drive TO the CNC WORD 1*)
D_Word2           : HLI_UNSI32; (*Data transmitted via the drive TO the CNC WORD 2*)
```

```
D_Word3      : HLI_UN32;(*Data transmitted via the drive TO the CNC WORD 3*)
D_Word4      : HLI_UN32;(*Data transmitted via the drive TO the CNC WORD 4*)
W_NativeDriveState : HLI_UN16;(*That's the original drivespecific state word from the drive
SERCOS, CAN ...*)
W_FillUp1    : HLI_UN16;
X_Mode0      : HLI_BOOLEAN;(*present mode 0F drive i.e. position loop control
DRIVE_STATE_MODE_0 0x00000001*)
X_Mode1      : HLI_BOOLEAN;(*DRIVE_STATE_MODE_1 0x00000002*)
X_Mode2      : HLI_BOOLEAN;(*DRIVE_STATE_MODE_2 0x00000003*)
X_Mode3      : HLI_BOOLEAN;(*DRIVE_STATE_MODE_3 0x00000004*)
X_Mode4      : HLI_BOOLEAN;(*DRIVE_STATE_MODE_4 0x00000005*)
X_Mode5      : HLI_BOOLEAN;(*DRIVE_STATE_MODE_5 0x00000016*)
X_Mode6      : HLI_BOOLEAN;(*DRIVE_STATE_MODE_6 0x00000007*)
X_ReadyPowerOn : HLI_BOOLEAN;(*Drive ready for Power ON DRIVE_STATE_READY_FOR_POWER_ON
0x00000008*)
X_ReadyControlLoopOn: HLI_BOOLEAN;(*Drive ready FOR Contol Loop ON
DRIVE_STATE_READY_FOR_CONTOL_LOOP_ON 0x00000010*)
X_Error      : HLI_BOOLEAN;(*Drive error DRIVE_STATE_ERROR 0x00000020*)
X_MeasureEquipActive: HLI_BOOLEAN;(*Messtaster (gauge) aktive DRIVE_STATE_GAUGE_ACTIVE
0x00000040*)
X_MeasureValueOk : HLI_BOOLEAN;(*Measure value is OK DRIVE_STATE_MEASURE_VALUE_OK
0x00000080*)
X_PitchErrorComp : HLI_BOOLEAN;(*pitch_error_compensation is active
DRIVE_STATE_PITCH_ERROR_COMP 0x00000100*)
X_FeedForward : HLI_BOOLEAN;(*feed_forward (Vorsteuerung)is active
DRIVE_STATE_FEED_FORWARD 0x00000200*)
X_BacklashErrorComp : HLI_BOOLEAN;(*backlash_error_compensation is active
DRIVE_STATE_BACKLASH_ERROR_COMP 0x00000400*)
X_DriftErrorComp : HLI_BOOLEAN;(*drift_error_compensation is active
DRIVE_STATE_DRIFT_ERROR_COMP 0x00000800*)
X_TempErrorComp : HLI_BOOLEAN;(*temperature_error_compensation is active
DRIVE_STATE_TEMPERATURE_ERROR_COMP 0x00001000*)
X_IsSai      : HLI_BOOLEAN;(*Axis is SAI*)
X_FillUp3    : HLI_BOOLEAN;(*fill-up members FOR MOD 4 alignment*)
X_FillUp4    : HLI_BOOLEAN;(*fill-up members FOR MOD 4 alignment*)
TYP_10_TwincatBoxData : TYP_10_TwincatBox;(* Communication data for a Twincat-Box *)
TYP_10_TwincatDeviceData: TYP_10_TwincatDevice; (* Communication data for a Twincat-Device *)
```

六、通道设置

通道设置主要包括对于各个通道的使用设定，可设定多通道，通道报警设定，通道倍率设定、通道接口设定、NC 代码行号查询、手动设定等。

1) CNC 报警处理

```
ErrorHandler_Channel1 : ChannelError;
```

程序

```
nChan := 1;
```

```
ErrorHandler_Channel1(ChannelNr:= nChan);
```

程序注释：第一个通道报警处理，通道报警处理在底层的 library 中进行报警。

2) 倍率设定

```
HLI_SetChannelOverride (nChan      := nChan,
                        nOverride   := PLCFeedOverride * 10);
```

调用子程序，将倍率数值赋值到子程序中，对 HLI 接口进行修改。

3) bolock search 程序 NC 代码行号查询

程序：

```
BlockSearch_Channel1
```

```
(nChan      := nChan,
```

```
  bStartRead := PLCMachineMode[nChan].BlockSearch.bStartRead, (* Edge signal *)
```

```
  bStartWrite := PLCMachineMode[nChan].BlockSearch.bStartWrite, (*Edge signal *)
```

```
  BlockSearch := PLCMachineMode[nChan].BlockSearch);
```

```
CNCSystem.Channel[nChan].ActDistance := BlockSearch_Channel1.coveredDistance / 10.0;
```

```
CNCSystem.Channel[nChan].ActBlockCount :=
```

```
BlockSearch_Channel1.actualBlockCount_AferReset;
```

程序注释：

1) 这段程序与界面相关联，实时读取当前的 NC 代码执行位置即中断计数行以及当前行执行的百分比。

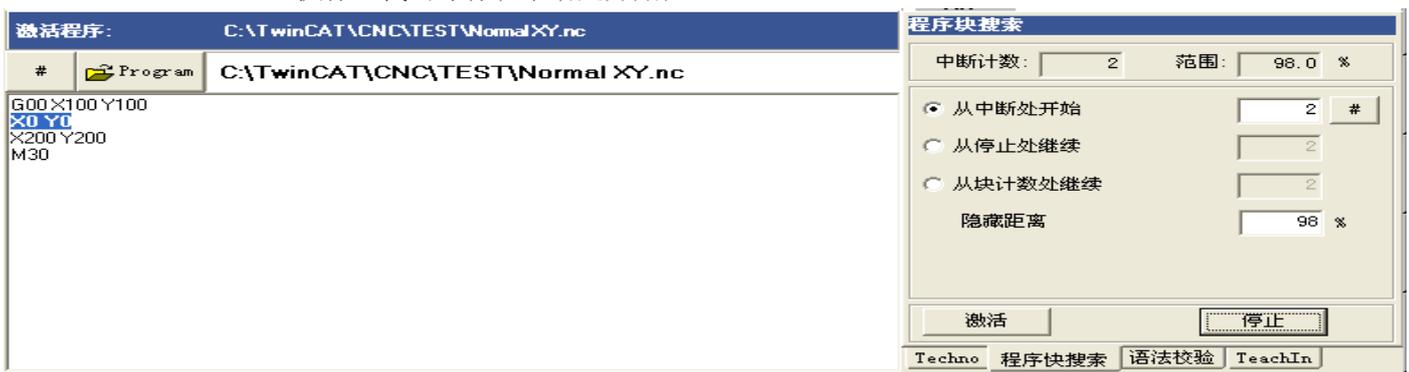
2) 中断计数类型

0 - 不执行 G 代码寻行

3 - 执行 G 代码寻行从块计数行开始；

4 - 执行 G 代码寻行从停止行开始；

5 - 执行 G 代码寻行从中断处开始；



4) 手动操作

程序:

```
ModeManual_Channel1( stManualModeParameter      := stManualModeParameter[nChan],
                    stManualModeControlElements := stManualModeControlElements[nChan],
                    bReset                      := FALSE,
                    stManualModeState           => stManualModeState[nChan]
                    );
```

程序注释: 手动程序都是通过这个子程序与 HLI 进行处理的, 与手动程序对应;

5) CNC 通道接口

程序:

```
CNCChannel1
(nChan          := nChan,
 (* Operation Mode *)
 bModeStandby   := PLCMachineMode[nChan].Standby,           (*待机模式*)
 bModeAutomatic := PLCMachineMode[nChan].Automatic,        (*自动模式*)
 bModeManual    := PLCMachineMode[nChan].Manual,           (*手动模式*)
 bModeMDI       := PLCMachineMode[nChan].MDI,              (*半自动模式*)
 bModeReference := PLCMachineMode[nChan].Reference,        (*回参模式*)
 (* Operation State *)
 bStateSelect   := bDummy,                                  (*选择*)
 bStateReady    := bDummy,                                  (*准备*)
 bStateStart    := bCycleStart,                             (*开始*)
 bStateStop     := bCycleStop,                              (*暂停*)
 bStateReset    := bCycleReset,                             (*复位*)
 (* Special Functions *)
 sProgramName   := PLCMachineMode[nChan].ProgramName,      (*调用 G 代码路径*)
 sMDIString     := PLCMachineMode[nChan].MDIString,        (*半自动 G 代码调用*)
 bBackwardMotion := PLCMachineMode[nChan].Backward,        (*回退功能*)
 bPrgBlockIgnore := PLCMachineMode[nChan].PrgBlockIgnore,  (*跳行功能*)
 bFeedHold      := PLCMachineMode[nChan].FeedHold,         (*停止使能*)
 bSingleBlock   := PLCMachineMode[nChan].SingleBlock,      (*单行执行功能*)
 bDoSingleBlock := PLCMachineMode[nChan].DoSingleBlock,    (*单步执行*)
 bM01Stop       := PLCMachineMode[nChan].M01Stop,          (*M01 暂停*)
 nSyntaxCheck   := PLCMachineMode[nChan].SyntaxCheck.nType (*语法检测*)
);
```

程序注释:

1) 模式相关

模式切换:

```

PLCMachineMode[1].Standby,          (*待机模式*)
PLCMachineMode[1].Automatic,        (*自动模式*)
PLCMachineMode[1].Manual,           (*手动模式*)
PLCMachineMode[1].MDI,              (*半自动模式*)
PLCMachineMode[1].Reference,        (*回参模式*)

```

模式状态:

```

CNCSysyem.Channel[1].ActMode=1-----等待-'StandbyMode'
CNCSysyem.Channel[1].ActMode=2-----自动-'AutomaticMode'
CNCSysyem.Channel[1].ActMode=3-----半自动-'MDIMode'
CNCSysyem.Channel[1].ActMode=4-----手动-'ManualMode'
CNCSysyem.Channel[1].ActMode=5-----回参-'ReferenceMode'

```

2) CNC 状态

CNC 系统当前状态:

```

CNCSysyem.Channel[1]. ActState =1-----Deselected
CNCSysyem.Channel[1]. ActState =2-----selected
CNCSysyem.Channel[1]. ActState =3-----ready
CNCSysyem.Channel[1]. ActState =4-----active
CNCSysyem.Channel[1].ActState = 5-----hold
CNCSysyem.Channel[1]. ActState =6-----error
CNCSysyem.Channel[1]. ActState =7-----'Selecting'
CNCSysyem.Channel[1]. ActState =8-----'Deselecting'
CNCSysyem.Channel[1]. ActState =9-----'Preparing'
CNCSysyem.Channel[1]. ActState =10-----'Clearing'
CNCSysyem.Channel[1]. ActState =11-----'Starting'
CNCSysyem.Channel[1]. ActState =12-----'Aborting'
CNCSysyem.Channel[1]. ActState =13-----ready
CNCSysyem.Channel[1]. ActState =14-----'Stopping'
CNCSysyem.Channel[1]. ActState =15-----resetting
CNCSysyem.Channel[1]. ActState =16-----'Finishing'

```

3) 操作相关

通过调用 HLI_CNCChannel 对 HLI 进行读写操作

```

bStateSelect    := bDummy,          (*选择*)
bStateReady     := bDummy,          (*准备*)
bStateStart     := bCycleStart,     (*开始*)
bStateStop      := bCycleStop,      (*暂停*)
bStateReset     := bCycleReset,     (*复位*)

```

4) 特殊功能

- G 代码调用调用地址

```
sProgramName      := PLCMachineMode[nChan].ProgramName,      (*调用 G 代码路径*)
指向 HLI:
```

- 等待模式下选择 G 代码路径

```
IF pChannel^.Mode.Standby = TRUE THEN
    HLI_ChannelModeStandby(nChan:=nChan, ProgramName:=sProgramName, pUnit:=pUnit,
        pCommand:=pCommand, pChannel:=pChannel);
END_IF
```

- 自动模式下选择 G 代码路径

```
IF pChannel^.Mode.Auto = TRUE THEN
    HLI_ChannelModeAuto(nChan:=nChan, ProgramName:=sProgramName, pUnit:=pUnit,
        pCommand:=pCommand, pChannel:=pChannel);
END_IF
```

- 回参模式下调用 G 代码路径，特殊一点事在回参模式下的 G 带默认为 rpf.nc 文件。

```
IF pChannel^.Mode.Reference THEN
    sProgramName := ''; (* we choose the default program: rpf.nc *)
    HLI_ChannelModeReference(nChan:=nChan, sRefPosString:=sProgramName, pUnit:=pUnit,
        pCommand:=pCommand, pChannel:=pChannel);
END_IF
```

- 半自动 G 代码调用地址

```
sMDIString        := PLCMachineMode[nChan].MDIString,        (*半自动 G 代码调用*)
```

指向 HLI

```
IF pChannel^.Mode.MDI THEN
    (*sProgramName := PLCCncMDIString;*)
    HLI_ChannelModeMDI(nChan:=nChan, pMDIString:=ADR(sMDIString), pUnit:=pUnit,
        pCommand:=pCommand, pChannel:=pChannel);
END_IF
```

- 回退功能

```
bBackwardMotion   := PLCMachineMode[nChan].Backward,        (*回退功能*)
```

指向 HLI

```
HLI_ChannelSetBackwardMotion(nChan:=nChan, bBackwardMotion:=bBackwardMotion);
```

- 单步执行功能:

```
bSingleBlock      := PLCMachineMode[nChan].SingleBlock,    (*单行执行功能*)
```

指向 HLI

```
HLI_ChannelSetSingleBlock(nChan:=nChan, bSingleBlock:=bSingleBlock);
```

单步执行功能与 HMIPRO 界面的单步运行功能相对应，协调使用。

■ 语法检测功能

nSyntaxCheck := PLCMachineMode[nChan].SyntaxCheck.nType (*语法检测*)

指向 HLI

HLI_ChannelSetSingleBlock(nChan:=nChan, bSingleBlock:=bSingleBlock);

语法检测功能包含虚拟运行、语法检测等功能，与 HMIPRO 界面配合使用，如下



选中相应的功能，单击‘激活’即选中相应功能，单击‘停止’即停止相应功能。选中功能后，单击开始，此时 G 代码执行相应功能。

相应功能如下：

0x0000	ISG_STANDARD	Normal mode
0x0001	SV	Block search
0x0002	SOLLKON	Offline Simulation with output of setpoint data
0x0802	SOLLKON	Offline Simulation without output of setpoint data
0x0004	ON_LINE	Simulation online visualisation
0x0008	SYNCHK	Simulation syntax check
0x0010	PROD_TIME	Simulation production time computation
0x0020	ONLINE_PROD_TIME	Simulation online production time computation
0x0040	MACHINE_LOCK	Dry run without axis movement
0x0080	TECHNO_FILTER	Dry run TECHNO filter
0x0100	KIN_TRAFO_OFF	Overwrites automatic enabling for kinematic transformations by a characteristic parameter defined in SDA_MDS (sda_mds*.lis).

涉及到的通道参数

```

pMC : ARRAY[1..HLI_SYS_CHNMAX] OF POINTER TO HLI_CHANNEL_ADDRESS;
HLI_CHANNEL_ADDRESS: addr : POINTER TO ChannelData;
ChannelData: StateBahn_Data : StateBahn;
pMC[1]^ . addr^ . StateBahn_Data.
D_CoveredDistance : HLI_SGN32; (*Gefahrener Weg im aktuellen Satz *)
D_CommandFeed : HLI_SGN32; (*programmierter Bahnvorschub*)
D_CommandFeedActive : HLI_SGN32; (*programmierter Bahnvorschub mit Echtzeiteinflussen*)
D_ActiveFeed : HLI_SGN32; (*aktueller Bahnvorschub*)
X_ProgramEnd : HLI_BOOLEAN; (*Programmende erreicht.*)
X_WaitErrorRemoval : HLI_BOOLEAN; (*int. Fehler - warten auf BF_ERR_CONTINUE*)
X_InterpolationActive : HLI_BOOLEAN; (*Interpolation ist aktiv*)
X_AxesInPosition : HLI_BOOLEAN; (*Achsgruppe in Position*)
X_WaitTechnoAcknowledge : HLI_BOOLEAN; (*Warten auf Quittierung der SPS*)
X_WaitContinue : HLI_BOOLEAN; (*int. Halt - warten auf BEWEG_FORTS*)
X_DwellTimeActive : HLI_BOOLEAN; (*Warten wegen Verweilzeit*)
X_WaitAxesInPosition : HLI_BOOLEAN; (*Warten auf Achsgruppe in Position.*)
X_BlockSearchActive : HLI_BOOLEAN; (*Satzvorlauf des Interpolators aktiv*)
X_SpeedLimitDetect : HLI_BOOLEAN; (*Unterschreitung der Grenzggeschwindigkeit*)
X_RapidMode : HLI_BOOLEAN; (*Eilgang aktiv*)
X_BlockInserted : HLI_BOOLEAN; (*Eingefuegter Satz *)
X_WaitContinueAfterCollision : HLI_BOOLEAN; (*warten auf BEWEG_FORTS kurz vor aufgetretener*)
X_Res1 : HLI_BOOLEAN; (*nicht verwendet *)
X_Res2 : HLI_BOOLEAN; (*nicht verwendet *)
X_Res3 : HLI_BOOLEAN; (*nicht verwendet *)
HLIBahnCoordDispData_Coord : ARRAY [1..GCW_250_HLIBahnBavoAxe] OF HLIBahnCoordDispData;
(*display data in actual coordinate system*)

HLIBahnCoordDispData:
HLIAxeName_Data : HLIAXeName;(*axis name*)
W_LogAchsNr : HLI_UN32;(*logical axis number*)
W_AxisType : HLI_UN32;(*axis type*)
W_State : HLI_UN32;
X_HomingDone : HLI_BOOLEAN;
X_FillUp_1 : HLI_BOOLEAN;
D_CmdPosition : HLI_SGN32;(*command position*)
D_CurrentPosition : HLI_SGN32;(*interpolated position*)
D_ActPosition : HLI_SGN32;(*actual position based on LR positions*)
D_ManPosition : HLI_SGN32;(*position incuding offset from maual mode*)
D_ManOffset : HLI_SGN32;(*offset from manual mode*)
D_DistToGo : HLI_SGN32;(*distance TO go*)
D_TotalOffset : HLI_SGN32;(* actual axis offset *)
D_AcsPosition : HLI_SGN32;(* ACS position *)
D_WOPosition : HLI_SGN32;(* w0 position, TCP position in base cartesian machine cs *)
AxeHBDisplayData_Data : AxeHBDisplayData;
HLISAddProgInfo_Data : HLISAddProgInfo;
D_StopConditions : HLI_UN32; (*aktuelle Haltebedingungen (Bit codiert)*)
D_BlockNumber : HLI_UN32; (*Aktuelle Satznummer*)

```

涉及程序框架部分包括如下子程序

功能: CNC→3 Control→ HLI_GetChannelError

该部分是对通道报警的读取;

功能: CNC→3 Control→ HLI_SetChannelOverride

该部分是设定 CNC 通道倍率;

功能: CNC→2 OperationMode→ HLI_CNCChannel

该部分是对 CNC 进行操作部分, 其中包括设定通道、功能开始、暂停、复位、G 代码执行路径、半自动 G 代码符号以及特殊功能回退、单步执行、语法检查等, 其中也对于 CNC 的功能进行了操作。

功能块: HLI_ChannelResetRequest

该部分执行请求状态等待复位;

特殊模式:

功能: CNC→3 Control→ HLI_ChannelSetBackwardMotion

该部分为回退功能设定;

功能: CNC→3 Control→ HLI_ChannelSetProgramBlockIgnore

该部分为通道设定忽略行;

功能: CNC→3 Control→ HLI_ChannelSetFeedHold

该部分为进给保持;

功能: CNC→3 Control→ HLI_ChannelSetSingleBlock

该部分为单行执行;

功能: CNC→3 Control→ HLI_ChannelSetReducedFeed

该部分为减少进给;

MHT 函数:

功能: CNC→4 MHT-Function→ HLI_ChannelResetMFunctions

该部分为对 M 函数进行初始化;

功能: CNC→4 MHT-Function→ HLI_ChannelResetHFunctions

该部分为对 H 函数进行初始化;

读取 HLI 信息:

功能: CNC→3 Control→ HLI_GetChannelStatusInfo

该部分为读取当前 HLI 的接口状态;

功能: CNC→3 Control→ HLI_GetChannelMode

该部分读取当前通道的模式;

功能: CNC→3 Control→ HLI_GetChannelState

该部分读取当前通道的状态;

功能块查找:

功能块: CNC→6 BlockSearch→ HLI_BlockSearch

该部分为功能块查找;



控制模式：

- 功能：CNC→2 OperationMode→ HLI_ChannelModeStandby
该部分待机模式下选择、准备、开始、停止、复位处理；
- 功能：CNC→2 OperationMode→ HLI_ChannelModeAuto
该部分自动模式下选择、准备、开始、停止、复位处理；
- 功能：CNC→2 OperationMode→ HLI_ChannelModeMDI
该部分半自动模式下选择、准备、开始、停止、复位处理；
- 功能：CNC→2 OperationMode→ HLI_ChannelModeReference
该部分回参模式下选择、准备、开始、停止、复位处理；
- 功能：CNC→2 OperationMode→ HLI_ChannelModeManual
该部分手动模式下选择、准备、开始、停止、复位处理；

控制复位：

- 功能：CNC→2 OperationMode→ HLI_ChannelReset
该部分通道复位；
- 功能：CNC→2 OperationMode→ HLI_AxisReset
该部分轴复位；

写入功能块序号：

- 功能块：CNC→6 BlockSearch→ WriteBsType
 - 功能块：CNC→6 BlockSearch→ WriteBsPass
 - 功能块：CNC→6 BlockSearch→ WriteBsCoveredDistance
 - 功能块：CNC→6 BlockSearch→ WriteBsAutoReturn
 - 功能块：CNC→6 BlockSearch→ WriteBsDeviationMax
 - 功能块：CNC→6 BlockSearch→ WriteBsBlockCount
 - 功能块：CNC→6 BlockSearch→ WriteBsBlockNumber
- 该部分为写入功能块；

读取功能块序号：

- 功能块：CNC→6 BlockSearch→ ReadBsPass
 - 功能块：CNC→6 BlockSearch→ ReadBsType
 - 功能块：CNC→6 BlockSearch→ ReadBlockCount
 - 功能块：CNC→6 BlockSearch→ ReadBlockNumber
 - 功能块：CNC→6 BlockSearch→ ReadBsAutoReturn
 - 功能块：CNC→6 BlockSearch→ ReadBsBlockCount
 - 功能块：CNC→6 BlockSearch→ ReadBsBlockNumber
 - 功能块：CNC→6 BlockSearch→ ReadBsCoveredDistance
 - 功能块：CNC→6 BlockSearch→ ReadBsDeviationMax
- 该部分读取当前执行行的序号以及百分比等。

手动模式:

功能块: CNC→2 OperationMode→ HLI_ChannelModeManual2
功能块: CNC→2 OperationMode→ PRG_AnalogJoystick
功能块: CNC→2 OperationMode→ HLI_SetAxisOverride
功能块: CNC→2 OperationMode→ HLI_AxisInManualMode
功能块: CNC→2 OperationMode→ HLI_ManualDeactivation
功能块: CNC→2 OperationMode→ HLI_ManualDeactivation
功能块: CNC→2 OperationMode→ HLI_ManualActivation
功能块: CNC→2 OperationMode→ HLI_ManualExecution
功能块: CNC→2 OperationMode→ HLI_ManualExecutionRapidMode

本章总结:

本章着重对于倍福 CNC 工程的 PLC 框架部分进行了总结,并以 MAIN 程序为总线进行讲解,涉及报警信息设定、创建以及显示。初始化程序的内容,以及轴、通道参数的 PLC 与 HLI 的交互以及原理,轴使能及涉及的功能块,手动操作模式、速度以及涉及的功能程序,通道报警、倍率设定、BlockSearch 功能、手动功能、CNC 通道接口如开始、暂停、复位、自动、手动、半自动、回参模式以及特殊功能回退、单步运行等功能。从本章可以对 CNC 框架有整体的认知,便于后续工作。

第六章 TwinCAT CNC 工程特殊功能介绍

概述

本章主旨在于用户自定义部分的功能的讲解，深入介绍一个功能的配置、PLC 编辑、NC 代码的应用等内容，对于客户现场开发具有重要作用。并且，本章将在后续不断完善，以增加功能的使用描述，便于学习、整理、记忆。

功能介绍

目前客户现场用到的功能以及特殊 G 代码使用包括回参（增量编码器、绝对值编码器）、龙门轴、反向间隙、螺距补偿、CNC 报警、搭配电机（SERCOS、模拟量、脉冲等）、scope 抓图等；

一、CNC 系统轴回参操作

参考点是机床的一个固定点，用于对机床运动进行检测盒控制的固定位置点，与机床原点的相对位置是固定的，机床出厂前的机床制造商精密测量确定。参考点位置有机械挡块或行程开关确定。数控系统通常要求机床回零操作，即使机床回到机床原点或机床参考点（不同的机床采用的会令操作方式可能会有所不同，但一般都要求回参考点，或者使用绝对之编码器）之后，才能操作机床。目的是使机床把运动部分和操作系统保持位置同步，只有机床参考点被确认后，刀具或工作台的移动才有基准。

倍福 CNC 轴回参分为虚拟轴回参、增量编码器回参、绝对值编码器回参三个部分，要实现回参需要通过 CNC 轴参数的配置、PLC 程序的调用以及 NC 代码的执行。

1、虚拟轴回参

虚拟轴情况下，回参不反向，不需要外部 IO，与真正的归原点是不同的。

参数：

P-AXIS-00161 rpf_weg_bis_nip 虚拟回参

注释：虚拟回参，从开始回参到回参结束的距离，这个参数仅用于虚拟回参，与其他回参方式不相关（虚拟归原点）。

2、增量编码器轴回参：

1) CNC 轴回参方式：

CNC 归原点方式分为三种

有 CAM sensor，有 ZERO PULSE，以 ZERO PULSE 归原点停下；

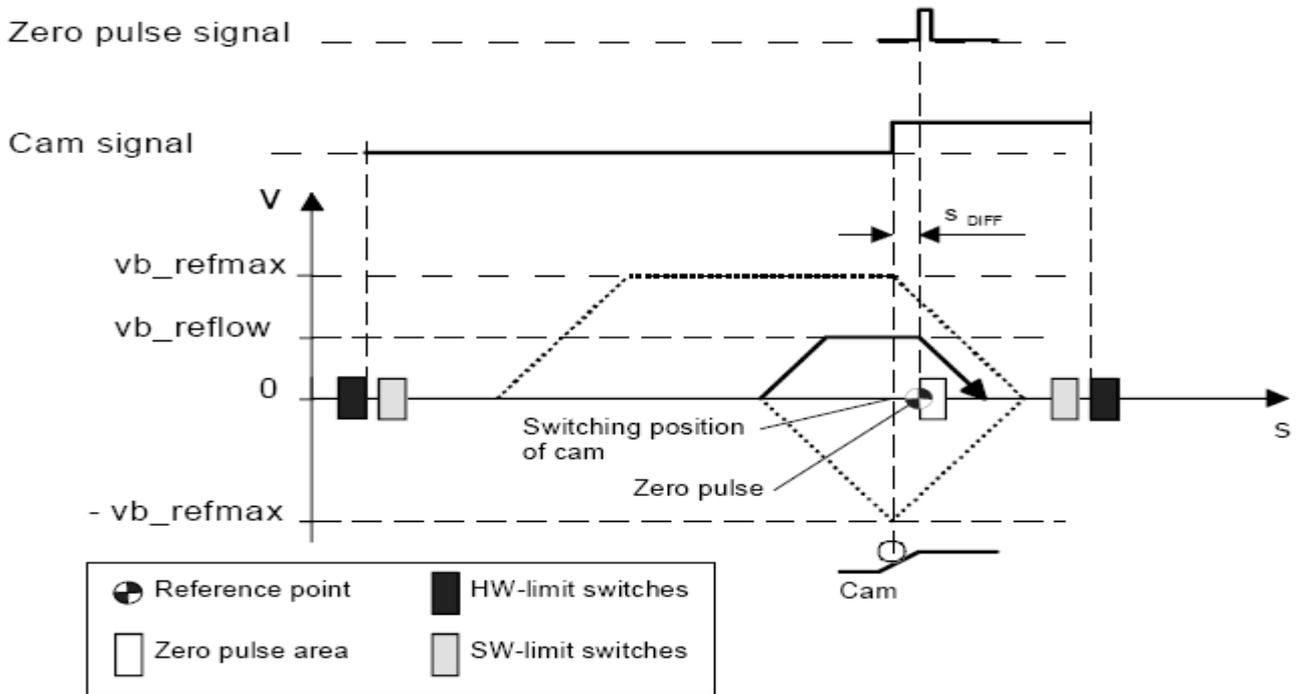
有 CAMsensor，没有 ZERO PULSE，以 CAM sesor 归原点停下；

无 CAMsensor，有 ZERO PULSE，以 ZERO PULSE 归原点停下；

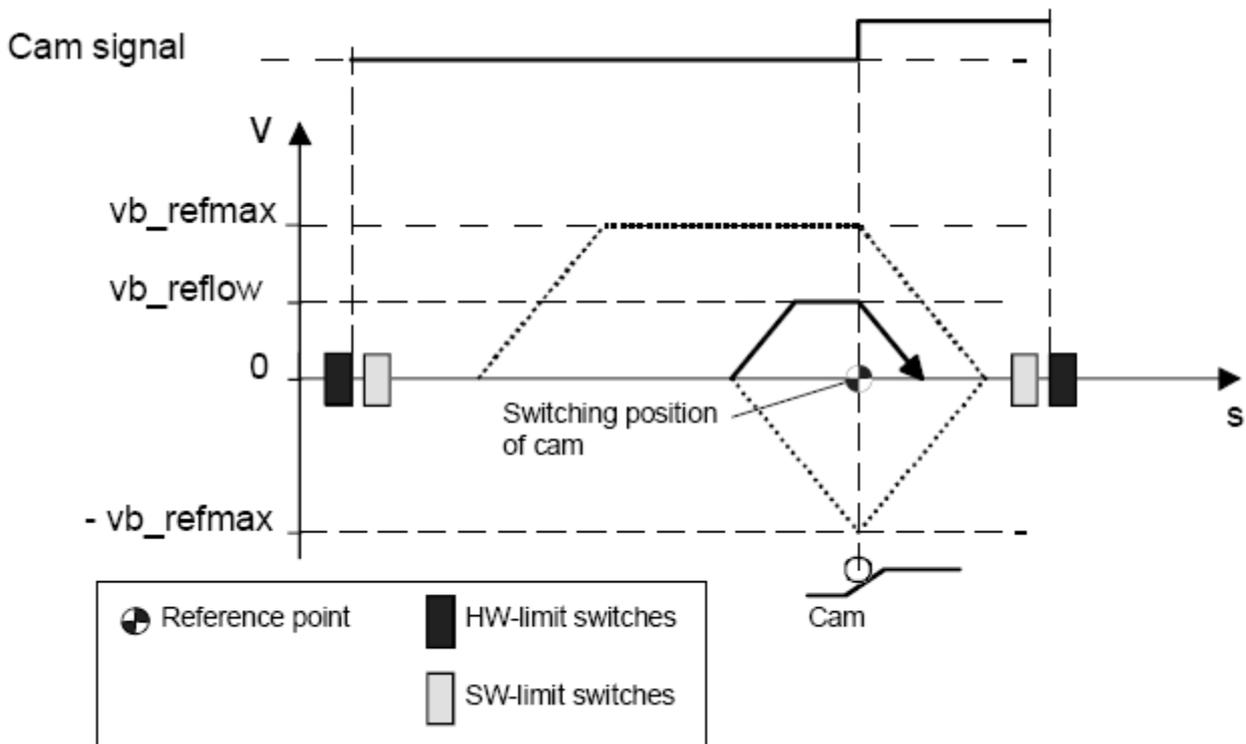
可以通过是否反向以及轴运动快慢速设定来实现多种归原点方式。

回参顺序如下：

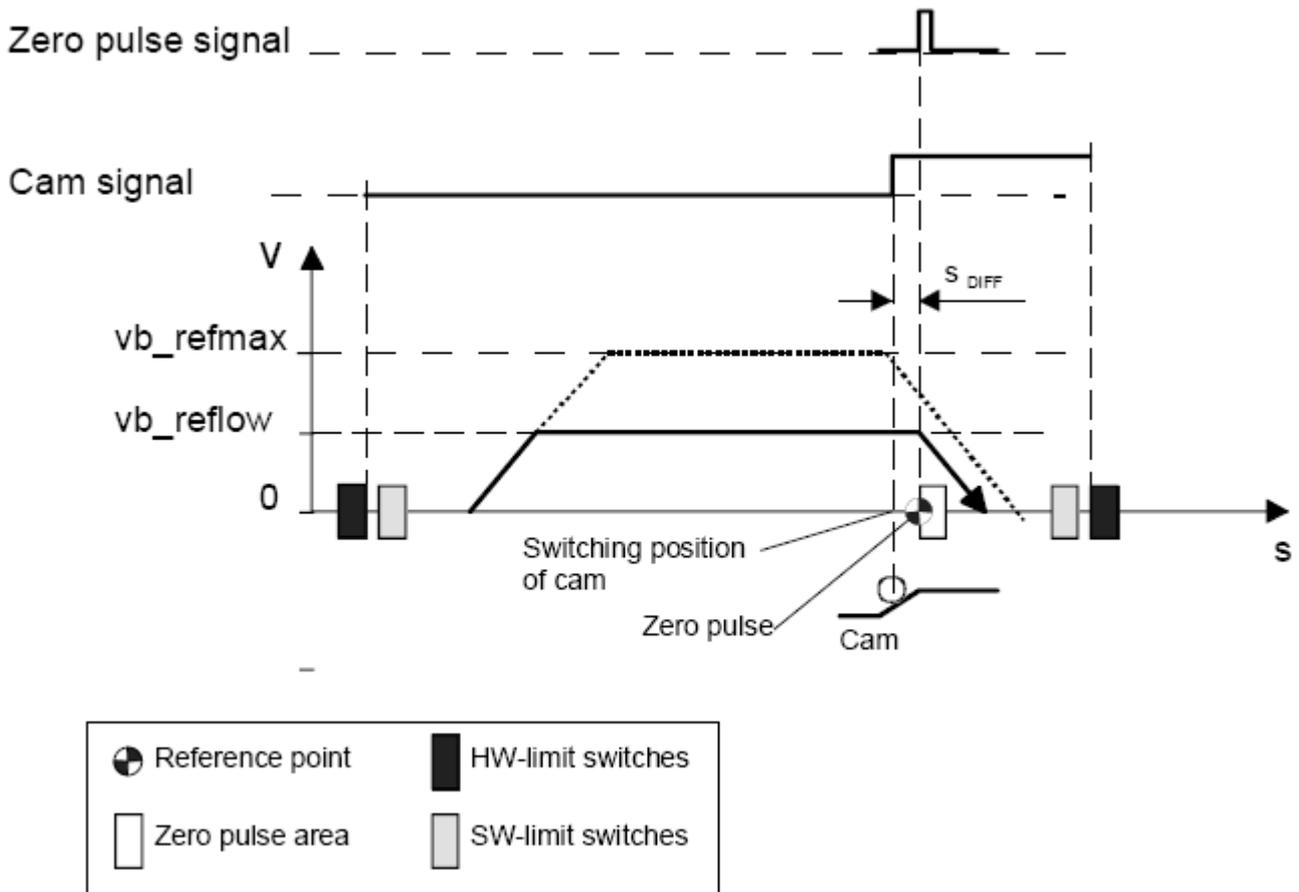
回参模式一：回参有零脉冲，有原点信号，反向，回参顺序如下：



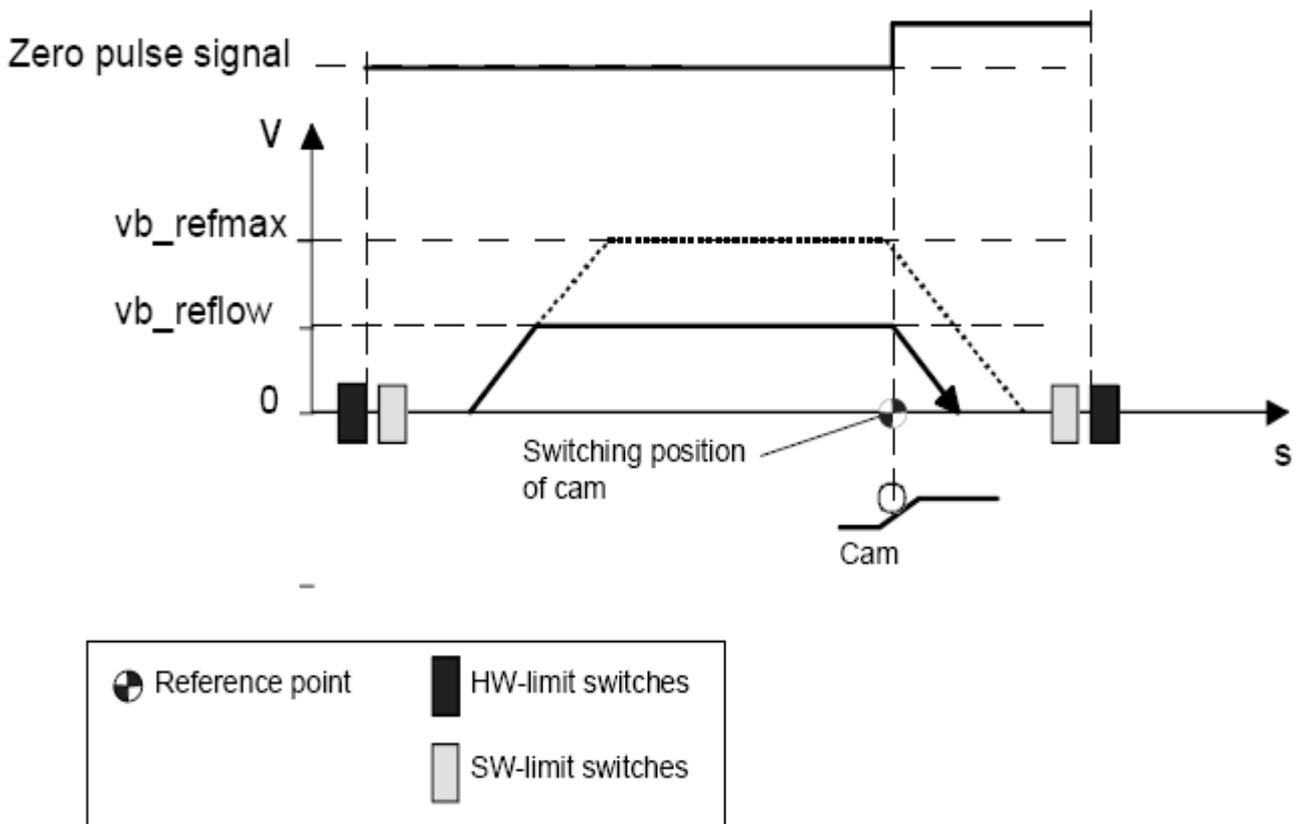
回参模式二：回参无零脉冲，有原点信号，反向，回参顺序如下：



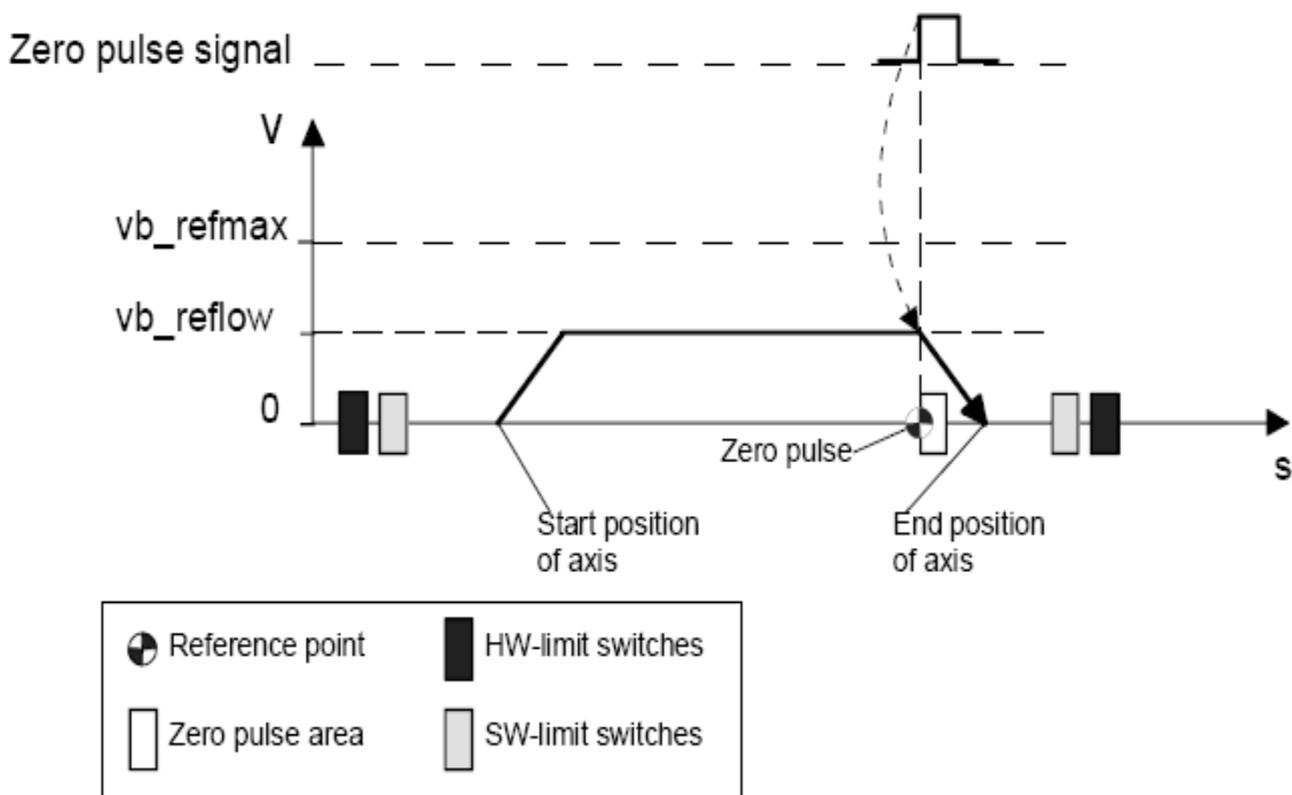
回参模式三：回参有零脉冲，有原点信号，不反向，回参顺序如下：



回参模式四、回参无零脉冲，有原点信号，不反向，回参顺序如下：



回参模式五：回参有零脉冲，无原点信号，不反向，回参顺序如下：



2) 实际配置轴参数设置：

➤ 轴类型设定

`kenngr.antr_typ` 2 # 驱动器类型。

注释：驱动器类型不需要设置，CNC 会根据当前的轴类型给出该值，

定义如下：

- 1: +/-10V drives with incremental encoders and use of a function library for I/O access (option)
- 2: SERCOS
- 3: Profidrive
- 4: Drive simulation (by digital filter)
- 5: Lightbus
- 6: Terminal (+/-10V drives with incremental encoders, transparent access via field buses)
- 7: Real-time Ethernet

➤ 回参时的加减速曲线设置

`getriebe[0].slope_type` 1 #回参过程中的加速度曲线类型

注释：该参数对两种情况生效，分别是 CNC 控制回参以及主轴运动。该参数具体定义如下

0: 阶跃响应；1: 梯形加速曲线；2: S 型加速曲线。

通常在应用中，加速曲线设定类型为 S 型曲线。

➤ 回参方向设定

`kenngr.ref_richt` 1 #回参时找参考点的轴运动方向

注释：设定回参方向 0: 负向，1: 正向。

4) 回参 NC 代码调用归原点

回参 G 代码调用，回参默认调用 G 代码名称 RPF.NC:

G 代码位置：与 ISG 路径调用一致，通常为 C:\TwinCAT\CNC\RPF.nc

G 代码回参示例：

```
G74 Z1 X2 Y2
```

注释：G74 是倍福 CNC 调用回参的代码，当前的程序表示 Z 轴先回参，之后 XY 再同时回参。

3、绝对值编码器轴回参

绝对值编码器不需要回参即可设定，设定包括配置文件以及 PLC 程序两部分。

1) 配置文件设定

- kenng.abs_pos_gueltig 0 # 绝对值编码器设定

注释：当驱动器编码器为绝对值编码器时，可设定为 1，不需要回参；绝对值编码器设定为 0；

- kenng.set_refpos_mode OFFSET # 回参设定数值类型

注释：回参位置类型 1、ABSOLUT；类型 2：OFFSET；类型 3：PLC；类型 4：PLC_OFFSET。

在绝对值编码器时，使用 OFFSET 以及 PLC_OFFSET 方式。

使用 OFFSET 方式，轴位置 = 编码器位置 + P-AXIS-00279 (kenng.set_refpos_offset)；

- kenng.set_refpos_offset -80611077 #[0.1um]偏置位置

注释在使用 OFFSET 方式下，这个值用于与编码器值相加，将轴位置设为 0；

注意：在使用 OFFSET 方式下，将 pos_refpkt 值设为 0，否则这个值将叠加到 OFFSET 中。

2) PLC 程序设定

将绝对值编码器轴，自动定义为回参完成，这样就可以正常运行。下例以第一根轴为绝对值编码器轴，PLC 程序如下：

```
pAc[1]^addr^.McControlLr_Data.MCControlBoolUnit_SetReferencePosition.X_Enable:= TRUE;
IF NOTpAc[1]^addr^.McControlLr_Data.MCControlBoolUnit_SetReferencePosition.X_StateTHEN
    pAc[1]^addr^.McControlLr_Data.MCControlBoolUnit_SetReferencePosition.X_Command:= TRUE;
END_IF
```

4、轴参数

ID	Parameter	Description
P-AXIS-00015	achs_mode	(轴模式选择)
P-AXIS-00014	abs_pos_gueltig	(绝对编码器OR增量编码器)
P-AXIS-00036	cam_direct_access	(0为HLI变量读入原点信号)
P-AXIS-00038	cam_level	(原点信号的类型, 0或者1)
P-AXIS-00064	fast_from_cam	(离开原点时速度选择快慢速)
P-AXIS-00074	gantry_slave_no_homing	(gantry从轴回参)
P-AXIS-00084	homing_without_zero_pulse	(回零是否0为使用0脉冲, 1为硬件CAM)
P-AXIS-00152	pos_refpkt	(设置归原点后的原点数值)
P-AXIS-00156	ref_ohne_nocken	(回参后是否使用CAM)
P-AXIS-00157	ref_ohne_rev	(归原点过程中反向回参: 0: 反向; 1: 不反向)
P-AXIS-00158	ref_richt Travel	(归原点运动方向)
P-AXIS-00161	rpf_weg_bis_nip	(虚拟回参距离归原点0.1um)
P-AXIS-00218	vb_reflow	(归原点慢速速度设定)
P-AXIS-00219	vb_refmax	(归原点快速速度设定)
P-AXIS-00299	homing_type	(CNC或者PLC归原点方式选择)
P-AXIS-00321	reference_cam_signal	Input of reference cam signal (only SERCOS)
P-AXIS-00321	kenngr.antr_typ	2 #Drive Interface SERCOS, 驱动器类型。2 Sercos, 8 canopen

本节总结:

本节主要对于回参方式做了讲解, 从回参的动作类型到回参的编码器类型选择以及轴参数设定如回参方向、回参速度、回参后位置等, 在 PLC 部分对于回参的 PLC 程序、回参功能启动以及 CNC 代码的执行方式都做了讲解。

二、CNC 系统龙门轴设定

概述：

龙门轴（GANTRY）主要用于处理大尺寸的龙门框在两边导轨上的移动，在移动过程中保持同步并且无偏差。在同步驱动运动指令时，一个轴接受运动指令运动同时另一个轴或者多个轴同步运动。接受移动指令的轴成为主动轴（Master Axis），与主动轴同步移动的轴成为从动轴（Synchronized Axis）。

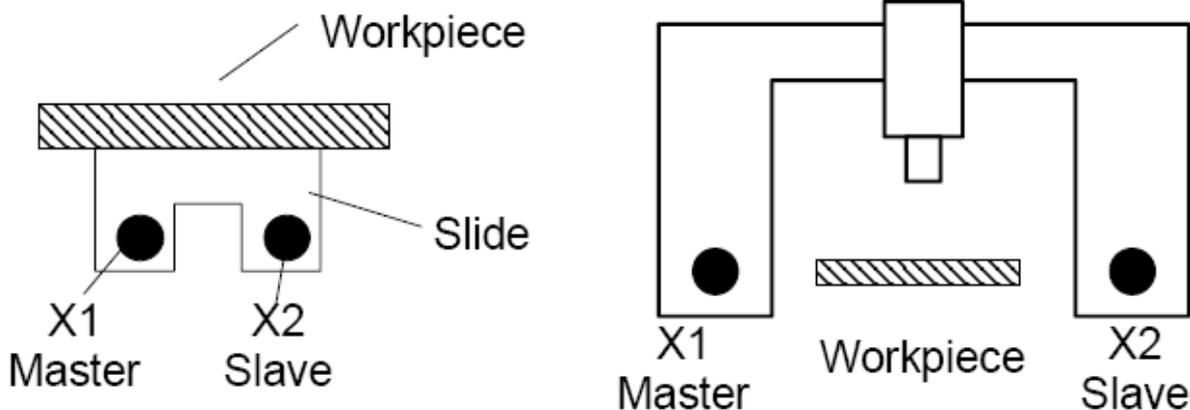
龙门轴说明：

倍福 CNC 系统的 GANTRY 轴分为 Soft Gantry(或者是 Coupled Axis)和 Mechanical Gantry(或者是 HARD Gantry)。Soft Gantry 用于不是刚性连接的两根轴（或者更多）的同步运动，通过程序宏指令来编辑，使用灵活。适用于正常情况下各轴相互独立，但有时候需要同时运动的情况。Hard Gantry 适用于两根轴在机械上刚性连接的龙门结构。通过轴参数设定为 Gnatry Axis，实现对于轴偏差进行监控，精度高。

1、Mechanical Gantry 设置：

1) 示意图：

此时主从轴为刚性机械结构，要求同步并且精度高，需要使用 Mechanical Gantry 设置。



2) 性能特点：

在开机、手动、手动回参考点、MDI 和自动等各种模式下都能实现精确同步运动，对主、从动轴的位置偏差一直进行监控，并在外界强力干扰以致轴对位置偏差过大、超出极限时能报警、停机，以防止机械损坏。

- 同步激活

由于机械的刚性联接，同步驱动在机床开机时，一直处于同步激活状态。

- 同步运动模式

所有运动指令在 JOG、手轮、增量运动、MDI 和自动等各种方式下，发送到主动轴的指令都将使从动轴同步移动。倍福的 HMIPRO 屏幕上不显示实际位置，也不能指令运动，在编程中从动轴应不出现。

- **偏差检测与报警**

在同步运动中,控制系统会一直检查两个或多个主从伺服电机的位置偏差,如果超过轴参数 P-AXIS-00072 设置数值时报警,此时可使用 reset 键清除报警;如果超过 P-AXIS-00071 设置数值时报警,此时使用 reset 按键也无法消除报警,只能通过重启 TWINCAT 的方式清除报警,按照情况进行修正两轴偏差。

- **归原点**

倍福的 CNC 可以实现两种方式归原点,分别为 CNC_Controlled 和 Driver_Controlled。

在 CNC_Controlled 控制下回参方式下,主轴回参与普通的 CNC 轴回参相同,主轴回参到位后将回参后的数值填写到 Gantry 从轴中,就认为此时主从轴回参完成。

在 Driver_Controlled 回参方式下,归原点时必须保证同时启动所有的 Gantry 轴;并且要求驱动归原点必须要执行同一段距离。

- **补偿功能**

补偿距离: 同步轴组中的主动轴和从动轴的补偿参数 Static difference 设置,设置在参数 P-AXIS-00073 kenngn.gantry_offset 补偿数值 OFFSET=SLAVE-MASTER

补偿速度设定: 从轴执行补偿距离的速度设定 P-AXIS-00075 kenngn.gantry_vb_korr

3) 轴参数设置:

Mechanical Gantry 的设置是通过轴参数直接设定的,不要 NC 代码设定,参数如下:

ID	Parameter	Description	
P-AXIS-00003	a_emergency	Axis acceleration in an emergency	轴急停加速度
P-AXIS-00015	achs_mode	Mode of an axis	轴模式
P-AXIS-00070	gantry_ax_nr	Axis number of the master axis	主轴序号
P-AXIS-00071	gantry_max_diff_reset_locked	Maximum distance difference	不可消除误差
P-AXIS-00072	gantry_max_diff_resetable	Distance difference that can be reset	可消除误差
P-AXIS-00073	gantry_offset	Static offset	机械偏置
P-AXIS-00074	gantry_slave_no_homing	Suppress reference point travel for gantry slave axis	龙门从轴归原点
P-AXIS-00075	gantry_vb_korr	Correction speed	纠偏速度设置
P-AXIS-00249	gantry_diff_check_without_homing	Monitor gantry difference before reference point travel	归原点前监控
P-AXIS-00253	gantry_synchronous_slave_homing	Drive-controlled reference travel of the gantry network	伺服归原点
P-AXIS-00254	cnc_controlled_stop_after_error	CNC-controlled error reaction	错误反馈位置

具体轴参数设置:

- **P-AXIS-00003:** a_emergency 急停时加减速速度设定。轴参数列表设置单位: mm/s²或者 °/s²。
- **P-AXIS-00015:** kenngn.achs_mode
如果是主轴: 0X10001
如果是从轴: 0X20001
最高位的 2 或者 1, 用来表示是从轴还是主轴, 最后一位用来表示是直线轴还是 360 度的旋转轴
- **P-AXIS-00070:** kenngn.gantry_ax_nr 从轴中设置的主轴序号
此参数设置在从轴轴参数中, 一旦设定此参数, 每次启动系统都将自动识别 Gantry 轴。

- P-AXIS-00071: `kenngr.gantry_max_diff_reset_locked` 主从轴之间最大的差值，超过此值将不能通过 `reset` 按键消除，只能重新启动系统或者 TWINCAT。
轴参数列表设置单位：0.1um 或者 0.0001°
Messer 值为：120000
- P-AXIS-00072: `kenngr.gantry_max_diff_resetable` 主从轴之间报警的界限差值，可以通过 `reset` 键消除。
轴参数列表设置单位：0.1um 或者 0.0001°
Messer 值为：62500
- P-AXIS-00073 `Static offset`
这个偏置值是主从轴之间的偏差值，其计算公式： $OFFSET = SLAVE - MASTER$ ；
轴参数列表设置单位：0.1um 或者 0.0001°。系统初始置为 0；
- P-AXIS-00074 `kenngr.gantry_slave_no_homing`
龙门轴从轴归原点是否执行；
通常为 1，即从轴不回参；
- P-AXIS-00075 `kenngr.gantry_vb_korr`
龙门轴从轴执行主从轴偏差距离的纠偏速度设置；
轴参数列表设置单位：1um/s 或者 0.001° /s；
- P-AXIS-00249 `kenngr.gantry_diff_check_without_homing`
控制系统是否在归原点执行前监视主从轴差值报警。
默认为在主从轴归原点之前不监视主从轴的差值报警
通常值为 1，控制系统在回参之前就监控报警。
- P-AXIS-00253 `kenngr.gantry_synchronous_slave_homing`
使用驱动控制归原点的启动各个轴的方式；
通常为 0，即从轴不回参；
- P-AIXS-254 `cnc_controlled_stop_after_error`
控制系统当 `gantry` 轴报警时的处理设置。
通常为 1，即出现错误时 CNC 控制停止。

2、SOFT GANTRY 设置:

1) 示意图:

此时主从轴是相互独立的，共同承载一个工件。此种结构适用于同步但精度不要特别高，使用SOFT GANTRY设置。

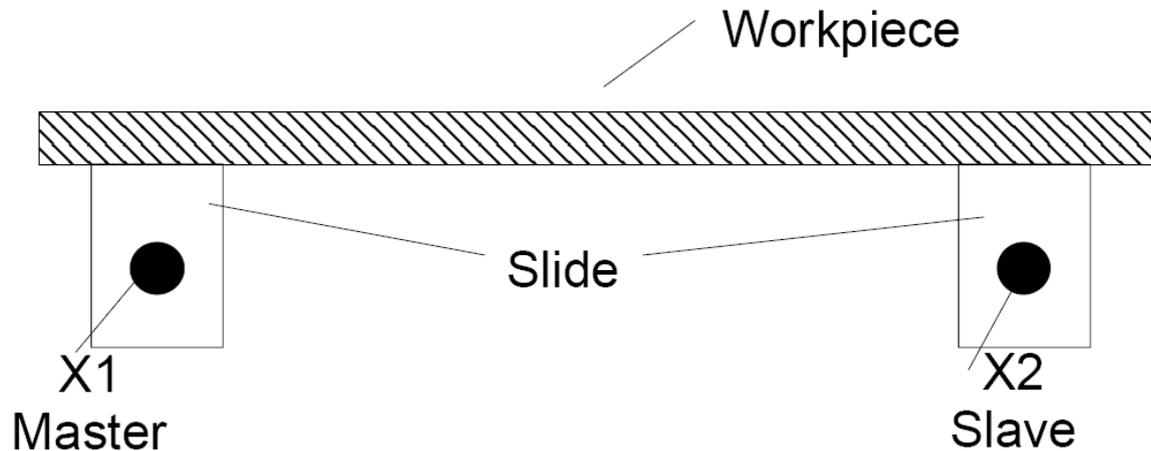


Figure 1-2: Programmable gantry operation ("soft" gantry)

2) 性能特点:

在实际的应用中，有一些龙门轴并不是机械结构连接的，只是在必要的时候或者同步不是很精确的情况下启用，此时使用SOFT GANTRY。

- **宏命令调用**

通过宏命令开启SOFT GANTRY，不需要在轴参数中设置主从轴；

- **主从轴偏置**

主从轴在SOFT GANTRY的情况下，主从轴的OFFSET不需要配置，是在内部通过位置来计算所得；

- **激活设置**

系统启动时不能够同时激活SOFT GANTRY，必须在NC程序中激活。同时通过参数设定是否在reset或者程序结束之后重新启动GANTRY设置。

- **多个SOFT GANTRY设置**

控制系统可以同时启动多个SOFT GANTRY设置；

- **手动限制**

当主从轴设置为同步模式时，从轴不能手动操作，也不能在NC程序中被编程。

2) 轴参数设置:

SOFT GANTRY中的轴参数比较少，主要是启动设置以及报警设置等，如下

ID	Parameter	Description	
P-AXIS-00003	a_emergency	Axis acceleration in an emergency	轴急停加速度
P-AXIS-00071	gantry_max_diff_reset_locked	Maximum distance difference	不可消除误差报警设置
P-AXIS-00072	gantry_max_diff_resetable	Distance difference that can be reset	可消除误差报警设置
P-AXIS-00075	gantry_vb_korr	Correction speed	纠偏速度设置
P-AXIS-00104	restore_coupling_after_reset	Restore coupling after reset	Reset之后是否维持GANTRY
P-AXIS-00105	Preserve_coupling_after_prog_end	Restore coupling after program end	程序结束之后是否GANTRY

具体轴参数设置:

- **P-AXIS-00003:** a_emergency急停时加减速度设定。轴参数列表设置单位: mm/s²或者° /s²。
- **P-AXIS-00071:** kenng.gantry_max_diff_reset_locked 主从轴之间最大的差值，超过此值将不能通过 reset 按键消除，只能重新启动系统或者 TWINCAT。
轴参数列表设置单位: 0.1um 或者 0.0001°
Messer 值为: 120000
- **P-AXIS-00072** kenng.gantry_max_diff_resetable 主从轴之间报警的界限差值，可以通过 reset 键消除。轴参数列表设置单位: 0.1um 或者 0.0001°
Messer 值为: 62500
- **P-AXIS-00075** kenng.gantry_vb_korr
龙门轴从轴执行主从轴偏差距离的纠偏速度设置;
轴参数列表设置单位: 1um/s 或者 0.001° /s
- **P-AXIS-00104** restore_coupling_after_reset
SOFT GANTRY 轴在控制器执行 RESET 之后，是否保持 GANTRY 状态;
0: 当 RESET 之后，不保持 GANTRY 状态;
1: 当 RESET 之后，保持 GANTRY 状态;
- **P-AXIS-00105** Preserve_coupling_after_prog_end
SOFT GANTRY 轴在控制器执行 NC 程序结束之后，是否保持 GANTRY 状态;
0: 当执行 NC 程序结束之后，保持 GANTRY 状态;
1: 当执行 NC 程序结束之后，不保持 GANTRY 状态;

3) 宏指令使用:

- 定义 SOFT GANTRY 组:

轴名称使用方式

```
#SET AX LINK [ <coupling_group>, [ <slave>=<master>,G [,<limit_1>, limit_2>] ]
                {, [ <slave>=<master>,G [,<limit_1>, limit_2>] ] } ]
```

轴序号方式

```
#AX LINK [NBR] [ <coupling_group>, [ <slave>=<master>,G [,<limit_1>, limit_2>] ]
                {, [ <slave>=<master>,G [,<limit_1>, limit_2>] ] } ]
```

参数定义:

<coupling_group> 耦合组序号;

<slave> 耦合组中的从轴名;

<master>耦合组中的主轴名;

NBR 耦合组必须以轴序号命名主轴从轴;

G 耦合组按照GANTRY轴类型;

<limit_1> 第一段限位, 可通过RESET消除的报警段设置;

<limit_2> 第二段限位, 不可通过reset消除, 重启twincat消除报警, 手动纠正变差。

- 定义 Mirroring 或者 Scalling 组:

注意在此种情况下, 系统不监控主从轴之间的差值!

轴名称使用方式

```
#SET AX LINK [ <coupling_group>, [ <slave>=<master>,<numerator>, <denominator> ]
                {, [ <slave>=<master>,<numerator>, <denominator> ] } ]
```

轴序号使用方式

```
#AX LINK [NBR] [ <coupling_group>, [ <slave>=<master>,<numerator>, <denominator> ]
                {, [ <slave>=<master>,<numerator>, <denominator> ] } ]
```

参数定义:

<coupling_group> 耦合组序号;

<slave> 耦合组中的从轴名;

<master>耦合组中的主轴名;

NBR 耦合组必须以轴序号命名主轴从轴;

<numerator>, <denominator> 分子分母, 必须为整数。

这两个参数目的是计算主从轴运动的比例参数。下列为可应用的比例参数

-1 : mirror coupling

1 : standard coupling; equivalent to the previous syntax

0 : output of an error message

- 使用 **SOFT GANTRY** 组:

激活 SOFT GANTRY:

#ENABLE AX LINK [<coupling_group>]

or

#ENABLE AX LINK (*Coupling group 0, defined in the channel parameter list*)

or alternative

#AX LINK ON [<coupling_group>]

oder

#AX LINK ON (*Coupling group 0, defined in the channel parameter list*)

关闭 SOFT GANTRY:

#DISABLE AX LINK [<coupling_group >]

or

#DISABLE AX LINK (*Deselection of the latest activated coupling group*)

or alternative

#AX LINK OFF [<coupling_group >]

or

#AX LINK OFF (*Deselection of the latest activated coupling group*)

#AX LINK OFF ALL (*Deselection of all active coupling groups*)

- 使用举例:

主轴设定: X, Y, Z, C

从轴设定: Y_S, Z_S, C_S

(初始化程序)

```
%L UP_INIT_ACHS_KOPPL (子程序耦合组1)
```

```
N10 #SET AX LINK[1, Y_S=Y, Z_S=Z, C_S=C]
```

```
N20 M17
```

```
%L UP_WZ (工具改变程序)
```

```
N30 #DISABLE AX LINK
```

```
N40 G01 G90 Y1000 Z100 C0 Y_S=1000 Z_S=100 C_S=0
```

```
N50 T10 D10
```

```
:
```

```
N80 G01 G90 X20 Y20 Z40 C50 Y_S=20 Z_S=40 C_S=50
```

```
N90 #ENABLE AX LINK[1] N110 M17
```

```
%L UP1 (子程序)
```

```
N150 G01 G91 X10 Y10 Z-20 C90
```

```
N160 G02 X20 Y20 I10 J10
```

```
N170 LL UP_WZ
```

```
N180 G01 G91 X10 Y10 Z-20 C90
```

```
N190 G02 X20 Y20 I10 J10
```

```
N200 M17
```

主程序:

```
N300 G01 G91 X20 Y20 Z40 C50 Y_S=20 Z_S=40 C_S=50 F300
```

```
N310 #ENABLE AX LINK[1] (or #AX LINK ON[1])
```

```
N320 LL UP1
```

```
:
```

```
N400 #DISABLE AX LINK (or #AX LINK OFF)
```

```
N410 M30
```

三、CNC 系统 MH 函数调用

概述:

M 函数和 H 函数是 G 代码执行过程中需要用到的功能函数。对于 BECKHOFF 的 CNC 系统而言，M 函数包含一些特殊功能，（例如 M00 程序暂停，M17 子程序停止，M30 程序停止等），除此之外，在 G 代码运行中还可以通过 M 函数调用 PLC 中功能子程序。M 函数是 G 代码中最不可或缺的部分。H 函数在功能上与 M 函数相似，相对于 M 函数来说，H 函数在使用时有大量连续的 H 函数可以使用（没有特殊功能函数），方便进行参数选择等功能。H 函数还可以通过修改对应的 PLC 程序，使其完成不同的功能。

1、CNC 的配置参数说明:

1、M/H 函数在 PLC 程序中引用:

CNCSystem.Channel[nChan].M[Nr]（nChan:=通道序号，Nr:=M 函数序号）

当 G 代码执行到 M 函数时，PLC 中变量 CNCSystem.Channel[nChan].M[Nr]置为 TRUE。可以利用这个变量来执行一个功能子程序，当子程序执行结束时，将变量 CNCSystem.Channel[nChan].M[Nr]置为 OFF，此时 CNC 轴才能继续执行运动。

H 函数在 PLC 程序中没有像 M 函数一样的数组，但底层上是相同。G 代码执行到 H 函数时，执行 PLC 中的 HLI_DoChannelHFunction 功能块。H 代码的使用见 [例程四](#)。

2、M/H 函数的类型:

（M\H 函数的执行方式根据设定的类型不同而发生变化，函数类型可以单独设置，如下图。）

符号	数值	内容
NO_SYNCH	0x00000000	M 函数不向PLC输出
MOS	0x00000001	M 函数向 PLC 输出，但是不跟 PLC 同步（ 例程四 ）
MVS_SVS	0x00000002	M函数在CNC运动模块之前就向PLC输出，且M函数在PLC完成之后置off，之后CNC继续运动。
MVS_SNS	0x00000004	M函数在CNC运动模块之前就向PLC输出，M函数执行时间较短，则CNC会继续运行下一行，否则CNC会减速到0，M函数在PLC完成之后置off，CNC继续运动轴。
MNS_SNS	0x00000008	M函数在CNC运动模块运行之后向PLC输出，等M函数运行结束之后，CNC继续运动轴。
MNE_SNS	0x00000020	M函数在触发一个事件之后输出给PLC，等M 函数运行结束之后，CNC继续运动轴。
提前输出		
MEP_SVS	0x01000000	M函数在进行下一个轴运动之前设置的的距离就对PLC输出，在轴运动之前与PLC同步。提前的距离设置为插补后的合成轨迹，通过P-CHAN-00070 (pre_outp[i])设置（ 例程三 ）
MET_SVS	0x02000000	M函数在进行下一个轴运动之前设置的的时间就对PLC输出，在轴运动之前与PLC同步。提前的时间设置通过P-CHAN-00070 (pre_outp[i])设置（ 例程三 ）

3、典型的参数配置举例：

通过 System Manager 开启 M/H 函数。

在 SYSTEM MANAGER ---> CNC CONFIGARTION ---> CHANNEL ---> SDA PARA 中定义 M/H 函数，其中“#”是屏蔽 M/H 函数标志，需要开启 M/H 函数的时，将其编号前的#删除。

```
# M-Functions and synchronization types
# =====
#m_synch[40] (M 函数 40 没有被开启)      0x00000002      MVS_SVS (M 函数的类型)
m_synch[41]                               0x00000002      MVS_SVS
m_synch[42]                               0x00000002      MVS_SVS
m_synch[43]                               0x00000002      MVS_SVS
m_synch[44]                               0x00000002      MVS_SVS
m_synch[45]                               0x00000002      MVS_SVS
# Festlegung der H-Funktionen und Synchronisationsarten
# =====
#h_synch[0]                               0x00000002      MOV
```

G 代码使用 M 函数：

函数 M25 在 CNC 轴运动之前就向 PLC 输出，且 M 函数在 PLC 完成之后置 off，之后 CNC 继续进行轴运动；M30 是 G 代码程序运行的结束语句。

```
N20 G00 X25
N30 X50
N40 X75 M25 (M25 of the MVS_SVS type)
N50 G01 X100 F2000
N60 X125 Z100
M30
```

提前输出 M 函数：

通过通道参数 P-CHAN-00070 (pre_outp[i])设置距离或者时间。

System Manager 设置： 函数 M96 在到达同步位置之前的 10mm 处输出到 PLC；
函数 M97 在到达同步位置之前的 40ms 时输出到 PLC；

```
# Definition of M functions and synchronisation methods
# =====
m_synch[96]                               0x01000000      MEP_SVS (提前位置输出)
m_synch[97]                               0x02000000      MET_SVS (提前时间输出)
#
# Pre-output time/distance setting with MET_SVS, MEP_SVS
# =====
pre_outp[96] 100000 in 0.1 um
pre_outp[97] 40000 in us
```

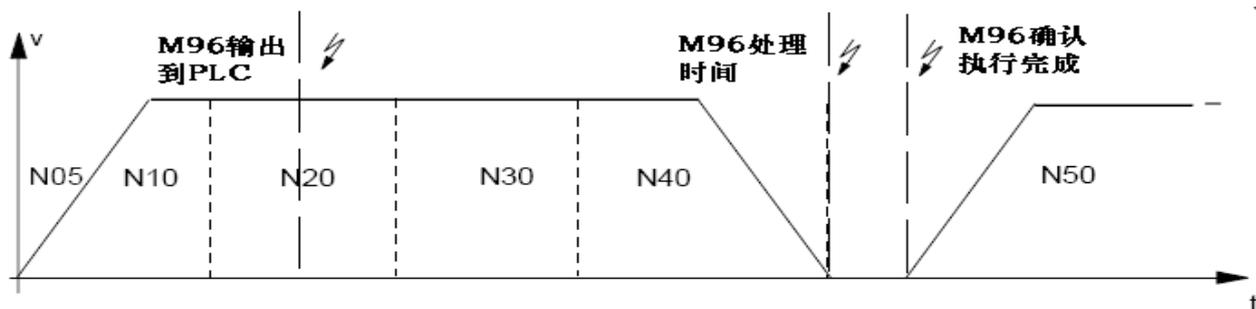
G 代码使用举例:

```

N10 G01 X10 G90 F5000
N20 X20
N30 X30
N40 X40
N50 M96 (M96 MEP_SVS pre_outp = 250000,) (or M97 MET_SVS pre_outp = 300000us)
N60 X80
N60 X0
M30

```

速度曲线如下图所示:



例程四：H 函数常规使用方法

在默认情况下，H 函数为 **MOS** 型，向 PLC 输出但不进行同步。当 G 代码运行到 H 函数时，向 PLC 发出信号，G 代码继续运行。当然也可以通过修改 PLC 程序中 HLI_DoChannelHFunction，PLC 功能块中设置对应 H 函数部分，如下：

```
CASE nChan OF (*nChan 为 CNC 通道号*)
```

```
1: (*检测通道 1*)
```

```
  CASE nFuncNr OF (* nFuncNr 为对应的 H 函数编号*)
```

```
    (*以下部分内容可以根据需要来进行添加修改*)
```

```
      0:      bSwith: =true; (*此处填写 H0 函数对应功能*)
```

```
            bBusy := FALSE;
```

```
      1:      bSwith: =true; (*此处填写 H1 函数对应功能*)
```

```
            bBusy := FALSE;
```

```
      2:      bBusy := FALSE; (*此处填写 H2 函数对应功能*)
```

```
    ELSE      (* 检测并退出 H 函数 *)
```

```
      ADSLOGDINT( msgCtrlMask := ADSLOG_MSGTYPE_MSGBOX OR ADSLOG_MSGTYPE_LOG,
```

```
      msgFmtStr := 'H-Funktion (%d), Channel 1, HLI_DoChannelHFunction() not implemented!',
```

```
      dintArg := nFuncNr);
```

```
      bBusy := FALSE;
```

```
    END_CASE;
```

```
  END_CASE;
```

G 代码使用举例:

```

N10 G01 X10 G90 F5000
N20 X100
N30 Y50
N40 X40
N50 H96 (执行 H 函数对应的功能，G 代码不停止，继续运行)
M02

```

四、CNC 系统 V.E 变量使用

概述:

在 NC 代码中，NC 代码中的数据可以与 GUI 以及外部 PLC 进行直接通讯获取变量数值的方式采用 VE 变量的方式。通过在配置文件中设置 VE 变量的名称、类型以及访问方式，以及设定一个内存区配置 VE 变量。VE 变量通过自启动配置文件的 ASCII 文件来激活外部变量。VE 变量可以方便的应用于 NC 代码中，实现条件判断或者执行轨迹运动。通过使用 index 的方式生成一片存储区域。这些变量可以生成和访问这片内存区域。

本节介绍 VE 变量的配置、NC 代码中使用以及 PLC 中通讯的相关操作编辑。实现一个 VE 变量的执行，可以分为三部分，包括配置文件中定义 VE 变量类型、数量等；PLC 程序中读写相应格式的外部变量，以及在 NC 代码中使用。

外部变量使用方式:

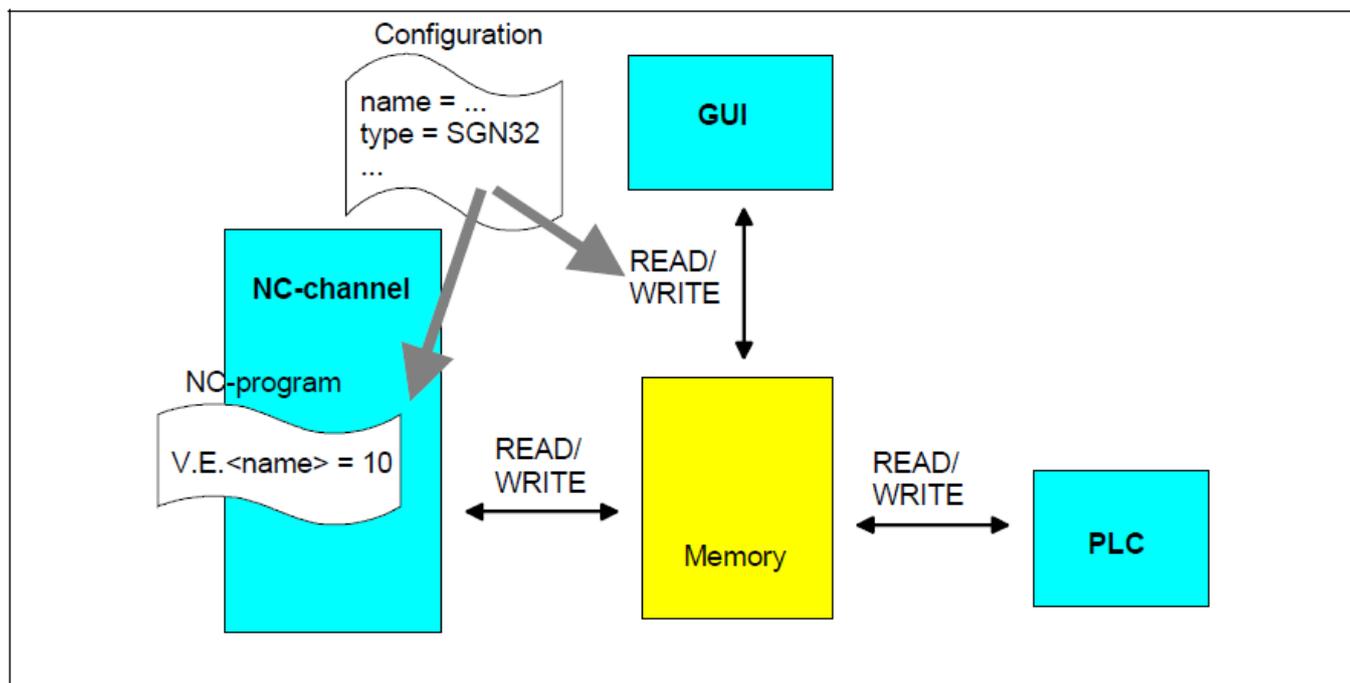


Fig. 1-1: Use of external variables

4.1 内存区域设定

每个通道的外部变量的内存区域必须在控制开始之前定义，对于TWinCAT系统来讲可以通过设定CNC Task-GEO中的HLI或者通过STUP中的变量PSTUP-00037 (ext_var_max)设定VE变量的内存区域。

4.2 配置 VE 变量

设置位置：通道项的 VE VAR 中设置，要求设置的参数 VAR[***]序号必须正确，不能出现间隔，便于与 PLC 中的 type 变量设置。

外部变量数量设定

anzahl_belegt 范围： [0; MAX_UN516]

当 var[i]中的[i]在整个配置文件中没有跳转，则这个值设定为 i+1;

当 var[i]中的[i]在整个配置文件中有跳转，则这个值设定为最大的数值 i*max+1;

外部变量设置参数

var[13].name	X_Homing_Position
var[13].index	3
var[13].type	REAL64
var[13].scope	GLOBAL
var[13].synchronisation	TRUE
var[13].access_rights	READ_WRITE
var[13].array_size	0
var[13].size	8
var[13].create_hmi_interface	0

注释: Var[i]: i 为 VE 变量的序列号。配置变量必须按照顺序从 0 开始,且中间最好不要有间隔;

变量	变量含义	注释
Var[i].name	VE 变量的名字	这个变量代表的是变量的名称,用于NC代码文件; 使用方式为V.E.<name>; 变量名成最大可以打动20个字符; 区分大小写;
Var[i].index	VE 变量的索引地址	当前变量在内存区域中的位置; 在 PLC 程序中,用于判断读写的变量判断;
Var[i].type	VE 变量的数据类型	设定值: [BOOLEAN,SGN08, UNS08,SGN16, UNS16, SGN32; UNS32,REAL64, STRING] 说明: 这个值设定当前变量的类型;
Var[i].scope	Scope 类型	设定值: [GLOBAL;CHANNEL] 说明: 用于判断全局生效还是通道生效;
Var[i].synchronisation	VE 变量的同步设定	设定值: [TRUE, FALSE] 说明: 通常设定读写访问与系统同步执行.
Var[i].access_right	VE 变量的读取权限	设定值: [READ_WRITE,READ_ONLY, WRITE_ONLY] 说明: 默认模式为读写,可以通过只读只写方式保护;
Var[i].array_size	VE 变量定义数组维数	数组定义,若不设定数组,此值设为 0
Var[i].size	VE 变量所占的数据长度	设定值: 1 BYTE-BOOLEAN,UNS08, SGN08 2-BYTE-SGN16, UNS16 4 BYTE-SGN32, UNS32, STRING 8 BYTE-REAL64 24 BYTE-STRING (maximum 20 characters) 根据变量的类型,设定数据长度
Var[i].create_hmi_interface	VE 变量在 hmi 接口设定	设定值: [TRUE, FALSE] 说明: 这个设定时用于生成界面接口;

数据长度定义:

数据类型	数据长度
BOOLEAN、SGN08、UNS08	1BYTE
SGN16、UNS16	2 BYTE
SGN32、UNS32	4 BYTE
REAL64	8 BYTE

4.3 PLC 程序使用

- 涉及子程序:

在 PLC 中读取 VE 变量数值子程序:

```
GetVEChannelBooleanValue  
GetVEChannelDintValue  
GetVEChannelIntValue  
GetVEChannelLrealValue  
GetVEChannelSintValue  
GetVEChannelStringValue  
GetVEChannelUdintValue  
GetVEChannelUintValue  
GetVEChannelUsintValue
```

在 PLC 中写入 VE 变量数值子程序:

```
SetVEChannelBooleanValue  
SetVEChannelDintValue  
SetVEChannelIntValue  
SetVEChannelLrealValue  
SetVEChannelSintValue  
SetVEChannelStringValue  
SetVEChannelUdintValue  
SetVEChannelUintValue  
SetVEChannelUsintValue
```

- 读取子程序:

```
VE_A3:=GetVEChannelBooleanValue(nChan :=1,nVEIndex:=2,nArrayIndex := 0,bGlobal:= TRUE);
```

nChan: 读取参数通道;

nVeIndex: 读取参数在 VE 变量中的 index 地址;

nArrayIndex: 读取参数在 VE 变量中的数组;

bGlobal: 读取标志位;

- 写入子程序:

```
SetVEChannelBooleanValue(nChan:=1,nVEIndex:=9,nArrayIndex:=0, val:=VE_A10,bGlobal:=TRUE);
```

nChan: 写入参数通道;

nVeIndex: 写入参数在 VE 变量中的 index 地址;

nArrayIndex: 写入参数在 VE 变量中的数组;

nVal: 写入参数的数值, 此处可使用变量编辑;

bGlobal: 写入标志位;

4.4 CNC 程序使用:

范例程序

```
$IF V.E.CHANNEL_WR >= 100 (In accordance with the value of)
                        (V.E.CHANNEL_WR branching to the)
                        (various cases occurs)

    G01 X100 Y100 F1000

$ELSE
    G01 X100 Y V.E.CHANNEL_WR F1000 (Linear interpolation in)
                                    (Y-direction with the value)
                                    (of V.E.CHANNEL_WR)

$ENDIF
    V.E.GLOBAL_SWR = V.A.ABS.X (The absolute X-coordinate is assigned)
                              (to the external variable)
    G01 XV.E.GLOBAL_SWR (Linear interpolation in X-direction)
                        (with the value of V.E.EXT2)
```

重点: V.E.+(配置文件中的响应变量地址);

五、CNC 系统中的区域保护功能

概述

机床在加工过程中，要求已不只局限在质量、效率、精度等方面，现在安全性已成为用户对机床要求中不可或缺的一部分。现在数控机床的发展速度迅猛，应用更加广泛，随之而来的一方面就是机床伺服轴的数量在不断增加，这样就出现了个别伺服轴安全工作区域因其他轴的移动，在不断变化的情况。因此只靠一些硬件限位开关来实现机床的行程保护就不可能了，必须利用系统中的软件来实现轴的行程保护及工作范围限定。本文说明倍福 CNC 系统对于工作区域不断变化的伺服轴保护的方法。

1、机床工作/保护区域

倍福 CNC 系统对于工作区域、保护区域限定非常灵活，这是一种软件保护的方法，可以在不同的控制轴的运动过程中，灵活修正。

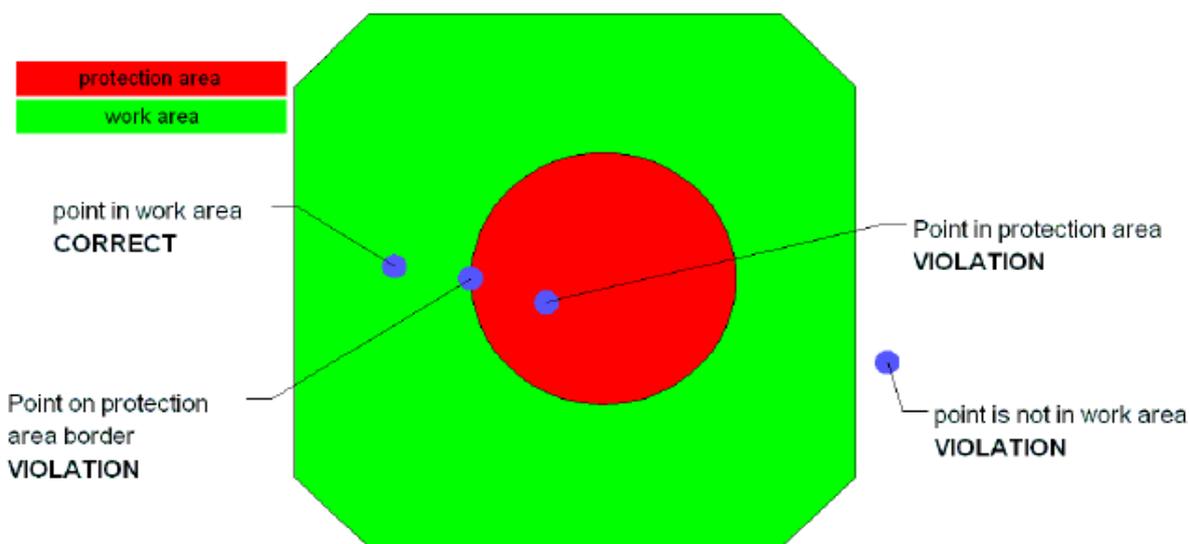


Fig. 1-2: Definition of work/protection areas

2、机床工作/保护区域实现机制

(一) 区域定义代码：

```
#CONTROL AREA
START | END----- (定义区域起始标志)
[
  ID<expr>----- (定义区域的代码)
  WORK | PROT ----- (定义区域的性质)
  POLY | CIRC----- (定义区域的类型)
  MIN_EXCUR<expr> X_EXCUR<expr>----- (定义区域的Z轴方向的上下限)
]
```

需要注意的是工作区域设定一定要实现一个完整的闭合线段，并且所有的位置必须为G90下的线段，使用机械坐标系。

(二) 工作区域定义:

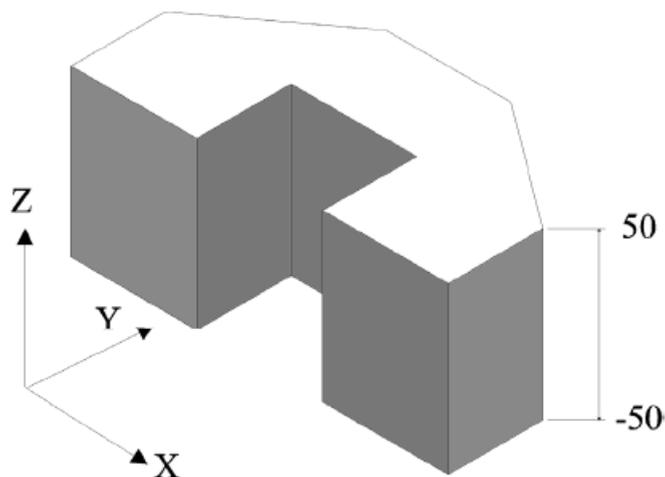
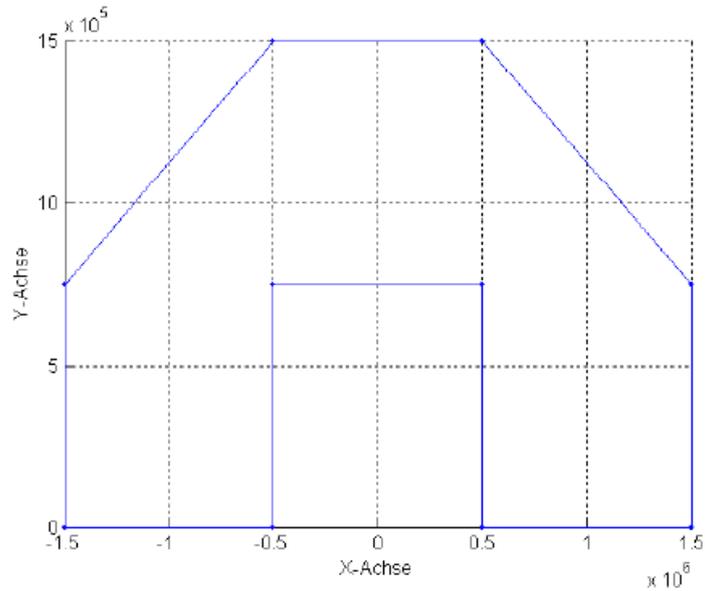
工作区域示例:

```

N10 #CONTROL AREA START [ID3 WORK POLY MIN_EXCUR=-50 MAX_EXCUR=50]
N20 G01 F1000 G90 X-150 Y75 (Start point)
N30 X-50 Y150
N40 X50 Y150
N50 X150 Y75
N60 X150 Y0
N70 X50 Y0
N80 X50 Y75
N90 X-50 Y75
N100 X-50 Y0
N120 X-150 Y0
N130 X-150 Y75 (End point identical with start point)
N140 #CONTROL AREA END

```

工作区域图:

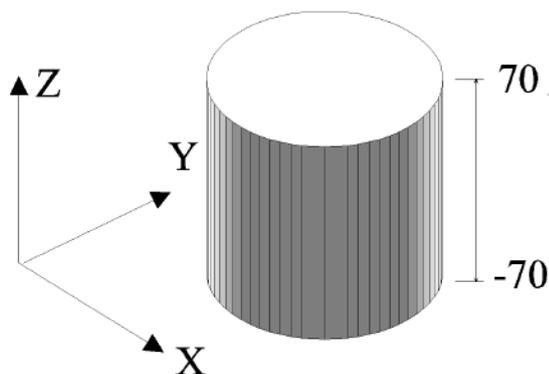
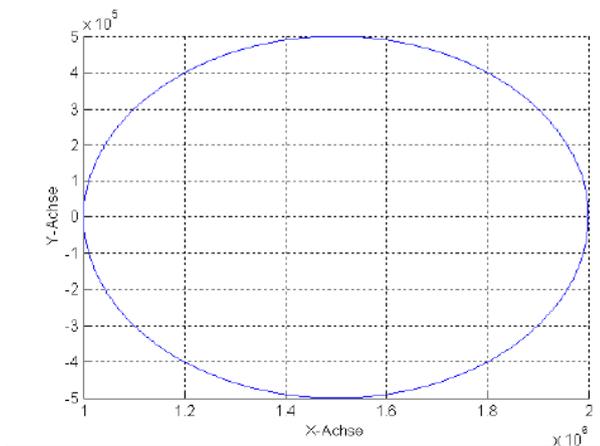


(三) 保护区:

```

N10 #CONTROL AREA START [ID4 PROT CIRC MIN_EXCUR=-70 MAX_EXCUR=70]
N20 G01 X100 Y0 F10000 (Start point for cyl. protection space)
N30 G02 G162 I50 J0
N40 #CONTROL AREA END

```



(四) 调用/关闭/清除区域:

- 调用工作/保护区

#CONTROL AREA ON [ID3]	(调用ID3工作/保护区)
#CONTROL AREA ON ALL	(调用所有定义的工作/保护区)
- 关闭工作/保护区

#CONTROL AREA OFF [ID3]	(关闭ID3工作/保护区)
#CONTROL AREA OFF	(关闭对应工作/保护区)
#CONTROL AREA OFF ALL	(关闭所有工作/保护区)
- 清除工作/保护区

#CONTROL AREA CLEAR [ID3]	(清除ID3工作/保护区)
#CONTROL AREA CLEAR ALL	(清除所有工作/保护区)

本章总结:

随着数控机床不断发展, 加工艺要求不断增加, 多轴机床产量的加大已是大势所趋。这些机床的安全性是生产厂家必须考虑的问题, 使用工作、保护区设定, 大大提高了工作的灵活性和安全性。

六、坐标系变换功能

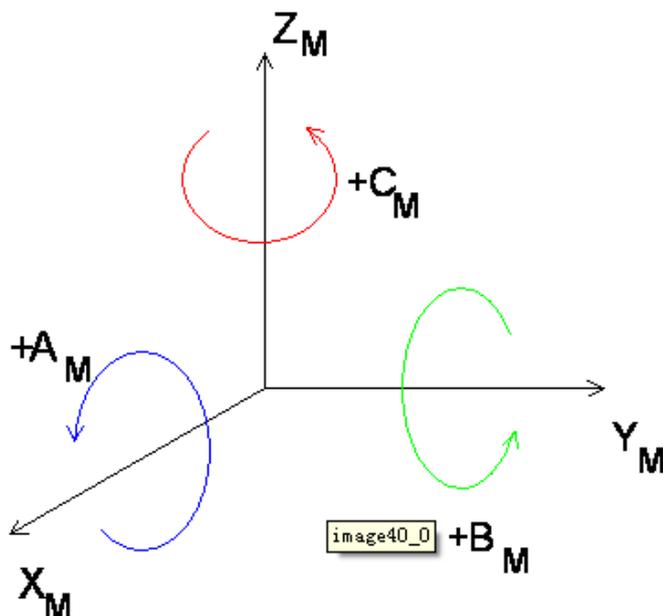
概述

机床的坐标系是机床工作区域的界定的依据。机床分为机械坐标系与工件坐标系，机械坐标系又可称为绝对坐标系，顾名思义，以机床的实际机械结构来衡量，通常是以参考点或限位来界定。而工件坐标系是建立在机械坐标系基础上的参考体系。在实际应用中，将机械坐标系转换为工件坐标系的方式有很多，各家的坐标系转换方式绝大多数是通用的，至少是在代码上是一致的，例如西门子、FANUC、Fagor 等于倍福都会采用 G53\G54 来命名机械坐标系。当然也存在的一些差异性，本文主要介绍倍福系统部分坐标系转换机制。

1、机床坐标系

机床坐标系

机床坐标系通常为 X、Y、Z、A、B、C，在此基础上进行坐标系变换。



2、机床坐标系变换实现机制

（一）位置偏移：XYZ 各轴位置偏移

- **G74 机械坐标系：**
 N10 G74 X1 Y2 Z3
 机床回参顺序：X→Y→Z

➤ **G53/G54..G59/G159 工件坐标系:**

G53-G59: 坐标系偏移数值设置在 channel→NP Para 中, 其中 G53 通常为坐标系偏置复位;

G159 = 0 对应 G53

G159 = 1 对应 G54

G159 = 2 对应 G55

G159 = 3 对应 G56

G159 = 4 对应 G57

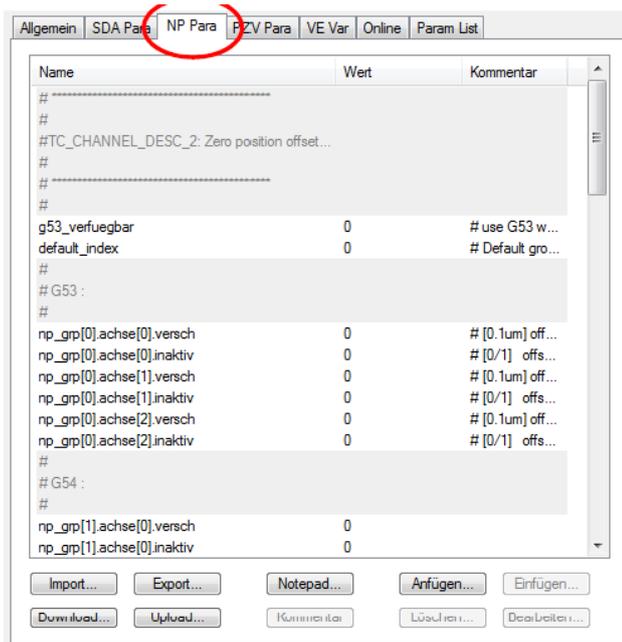
G159 = 5 对应 G58

G159 = 6 对应 G59

G159 = 7 对应 channel→NP Para 表格中的 index 号码为 7

:

G159 = 10 对应 channel→NP Para 表格中的 index 号码为 10



➤ **#PSET/#PRESET 工件坐标系**

N001 #PSET X0 Y0 (置当前位置为 X0 Y0)

N003 #PRESET (清除当前坐标系)

➤ **G92 工件坐标系**

N001 G92 X100 Y20 (设置当前坐标系原点为 X100 Y20)

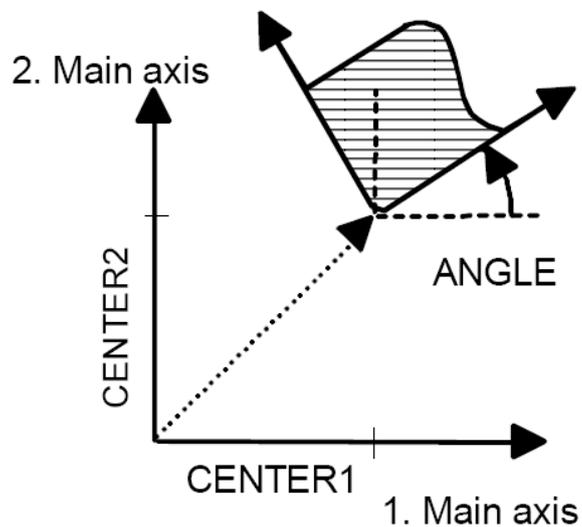
(二) 角度偏移: XYZ 各轴坐标系角度偏移

➤ **#ROTATION ON/OFF** 工件坐标系

```
#ROTATION ON [ANGLE -45 CENTER1=10 CENTER2=100]
```

(当前工件坐标系, 原点偏移至X10 Y100, 角度偏置为 -45°)

```
#ROTATION OFF (关闭坐标系偏移)
```



本章总结:

倍福 CNC 控制在坐标系变换中的形式很灵活, 对于终端用户来讲, 非常容易掌握, 是倍福的 CNC 控制系统的亮点。

七、测量探针功能

概述

探针测量功能用于机床需要实现 G 代码执行过程中的中断处理，在倍福 CNC 中，通常采用 G100/G310 功能函数实现。当触发探针测量功能时，轴的测量信息被保存在 CNC 共享内存区中，可以通过 VA 变量以及 PLC 读取。可以使用驱动器以及 PLC 变量实现探针信号的读入。使用探针功能需要配合设置相关的轴参数以及通道参数。

1、测量模式定义

测量模式共包括 7 中不同的模式，如下：

Measuring types	Significance
1	Measurement travel with at least one axis, Measurement feed programmable through F-Word.
2	Measurement travel with exactly one axis. Measurement feed is specified in the axis parameter list [2]-5.
3	Measurement travel with at least one axis, Measurement feed programmable through F-Word , optionally further traverse up to the destination point.
4	Measurement travel only with the maximum of 3 main axes, Measurement feed programmable through F-Word.
5	Interruptible measurement travel with at least one axis, Measurement feed programmable through F-Word.
6	Interruptible measurement travel with at least one SERCOS-Axis, Measurement feed programmable through F-Word.
7	Measurement travel with moving on a fixed stop with at least one axis, Measurement feed programmable through F-Word

2、测量模式选择

通过两种方式启动：

通道参数启动 P-CHAN-00057 （messtyp）默认的测量模式；

通过 G 代码宏指令启动 #MEAS MODE[VAR]，模式选择代码如下：

#MEAS MODE [[<expr>]]	(modal)
-----------------------	---------

<expr> Measurement type acc. to chapter 4.1.10.

The programming of #MEAS MODE without parameter selects the default measurement type, which is specified in the channel parameters list [1]-1.



Programming example

```

N10 #MEAS MODE[3]           (Measurement type 3)
N20 G100 X150               (Measurement traverse)
..
N100 #MEAS MODE             (Default- measurement type)
..
M30

```

3、测量模式说明

3.1 G100 类型一-----多轴测量

指令格式:

G100 <axis_name><expr> { <axis_name><expr> } [F<expr>]	(non-modal)
--	-------------

G100	Selection measuring traverse
<axis_name><expr>	Target point of measuring axis
F<expr>	Measuring feed

指令说明:

- 模式一设定时，允许所有轴进行测量功能。要求在各轴中对轴参与测量功能的参数进行设定。
- 测量程序执行时，速度值设定是在 F 指令中进行设定，程序的执行如 G01 是一致的，中间遇到测量信号时，停止执行当前目标位置，跳转至下一行执行。

指令举例:



Programming example

```

%Meas_traverse
N10 G00 X0 Y0 Z0
N20 X5
N30 G100 X10 Y10 F500
N40 G01 X7
N50 M30
  
```

Fig. 4.6 shows the representation of the resulting path:

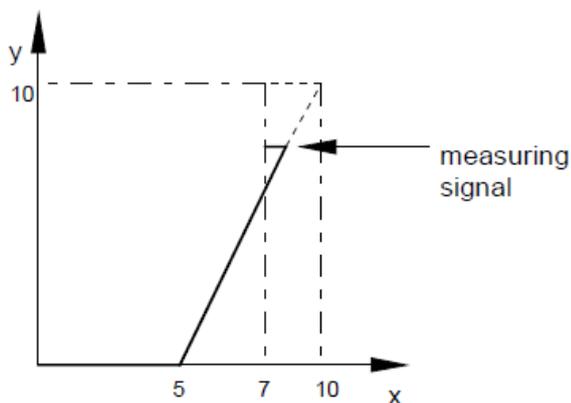


Fig. 4.6: Programmed path

3.2 G100 类型二-----单轴测量

指令格式:

G100 <axis_name><expr>	(non-modal)
-------------------------------	-------------

G100	Selection measuring traverse
<axis_name><expr>	Target point of measuring axis

指令说明:

- 单轴测量功能，需要在轴参数中将 `kenngr.messachse` 设定，并使用模式二；
- 测量功能执行时，代码执行速度由 `P-AXIS-000215` 参数进行设定，此时的 `F` 指令失效。执行过程中，采用 `G01` 的方式，遇到测量信号时，停止当前程序执行，跳转至下一个 `block` 动作。

指令举例:



Programming example

```
%Meas_traverse
N10 G00 X0 Y0 Z0
N20 Y5
N30 G100 X10
N40 G01 X7
N50 M30
```

Fig. 4.8 shows the representation of the resulting path:

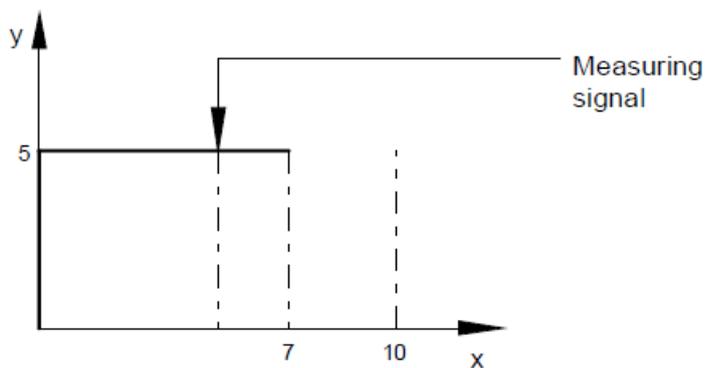


Fig. 4.8: Programmed path

3.3 G100 类型三----执行到目标位置

指令格式:

G100 <axis_name><expr> { <axis_name><expr> } [G106] [F<expr>]	(non-modal)
---	-------------

G100	Selection measuring traverse
<axis_name><expr>	Target position of measuring axis
G106	Movement to target point
F<expr>	Measuring feed

指令说明:

- 所有轴参与测量功能，轴参数设定是否参与多轴测量，同时使用类型三；
- 测量功能执行时，代码执行速度 F 指令指定。执行过程中，轨迹执行动作与 G01 动作类似；
- 检测到测量信号后，如果本行程序中包含有 G106 则执行到目标位置，如果不包含 G106 则停止本次目标位置的移动，开始执行下一个 block 行号。

指令举例:



Programming example

```

%Meas_traverse
N10 G00 X0 Y0 Z0
N20 G01 X5 F500
N30 G100 G106 X10 Y10 ;After measuring signal traverse to target point
N40 G01 X7
N50 M30

```

Fig. 4.10 shows the representation of the resulting path

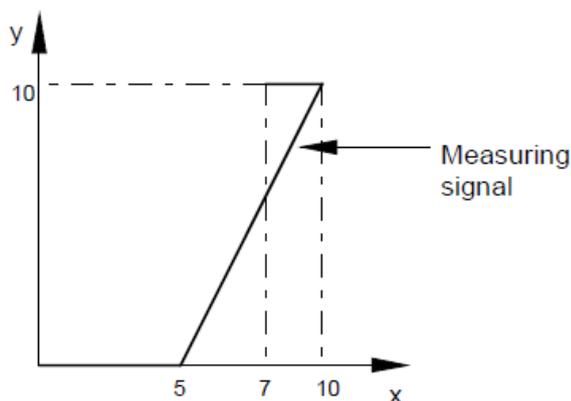


Fig. 4.10: Programmed path

3.4 G100 类型四----基础坐标轴测量功能

指令格式:

G100 <axis_name><expr> { <axis_name><expr> } [F<expr>]	(non-modal)
---	-------------

G100	Selection measuring traverse
<axis_name><expr>	Target position of measuring axis
F<expr>	Measuring feed

指令说明:

- 所有基础轴参与测量功能，轴参数设定是否参与基础轴测量，同时使用类型四；
- 测量功能执行时，代码执行速度 F 指令指定。执行过程中，轨迹执行动作与 G01 动作类似；

指令举例:



Programming example

```

%Meas_traverse

N10 G00 X0 Y0 Z0
N20 X5
N30 G100 X10 Y10 F500
N40 G01 X7
N50 M30
  
```

Fig. 4.12 shows the representation of the resulting path:

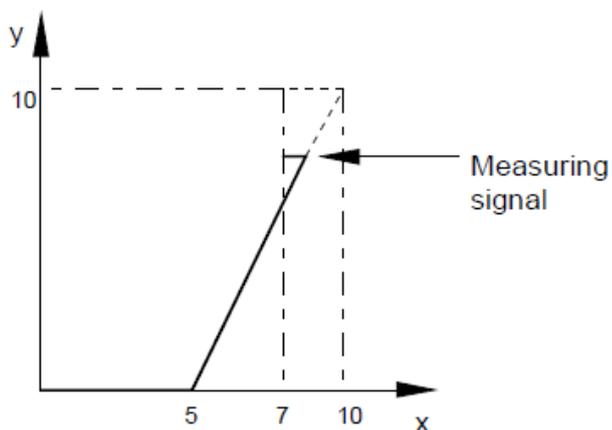


Fig. 4.12: Programmed path

3.5 G310 类型五、六

指令格式:

```
G310 [G00 | G01] <axis_name><expr> {<axis_name><expr>} [$GOTO<Label>] (non-modal)
```

G310	Interruptible block
G00 G01	Interruptible mode of interpolation
<axis_name><expr>	Target position of measuring axes
\$GOTO<Label>	Jump label after interrupted measuring traverse

指令说明:

- 所有基础轴参与测量功能，轴参数设定是否参与基础轴测量，同时使用类型四；
- 测量功能执行时，代码执行以设置的 G00/G01 指令并以速度 F 指令指定。执行过程中，当遇到检测信号后，停止执行当前程序，跳转到 G310 中设定的行号，如果在执行过程中，没有触发测量信号，则程序继续执行当前行。



Programming example

```
N10 G00 X0 Y0
N20 G310 G01 F100 X100 Y200 $GOTO[N_LABEL] (If interrupt. jump to N_LABEL)
N30 G01 X200
N40 $GOTO[ENDE]
N50 [N_LABEL] X0 Y0
N60 [ENDE] M30
```

3.6 G100 类型七

指令说明:

- 所有基础轴参与测量功能，各轴需要设定力矩限制，同时关闭位置监控功能；
- 龙门轴中，如论软龙门还是硬龙门结构，测量功能执行时，只是测量主轴功能，从轴只是跟随。此时的从轴同样需要设置力矩限制，并关闭位置监控功能；

3.7 与测量功能相关的轴、通道参数

ID	Parameter	Description
P-CHAN-00057	messtyp	Setting the default measurement type

ID	Parameter	Description
P-AXIS-00086	hub_messtaster	Measuring probe stroke
P-AXIS-00113	mess_neg_flanke	Measured signal edge
P-AXIS-00114	mess_offset	Permissible distance to destination point in case of measurement type 2
P-AXIS-00115	mess_signal_achs_steuer	Consideration of external measured signals
P-AXIS-00116	mess_signal_sercos	Reading in the measured signal in the case of SERCOS
P-AXIS-00117	mess_signal_taster	Measuring probe signal via hardware interface
P-AXIS-00118	messachse	Axis can be used as measured axis
P-AXIS-00215	vb_messen	Feed rate for measurement in accordance with measurement type 2
P-AXIS-00060	echtzeit_bit_nr	Number of used real time bits for SERCOS drives
P-AXIS-00257	probing_signal_via_plc	Measuring is done in the CNC
P-AXIS-00330	meas_signal_fixed_stop	Measuring with movement on a fixed stop
P-AXIS-00331	fixed_stop_pos_lag_limit	Limit for position lag
P-AXIS-00332	fixed_stop_nbr_cycles	Number of position controller cycles

3.8 与测量功能相关的 V.A 变量

V.A.MESS.<axis>

V.A.MESS.X: X的测量值;

V.A.MERF.<axis>

V.A.MERF.X: X轴测量功能是否执行 (TRUE/FALSE), 即X轴在执行测量功能时, 是否已经执行到目标位置之前出发了测量功能;

V.A.MOFFS.<axis>

V.A.MOFFS.X: 测量功能执行后, 测量信号点到目标位置点之间的距离;

V.A.MEIN.<axis>

V.A.MEIN.X: 测量偏置值, 即X轴实现测量功能时, 执行G101得到当前测量位置偏置。

$$PCS' = PCS + \text{measurementoffset}_{G101}$$

There is following relation between V.A.MESS.<name> and V.A.MOFFS.<name>:

	without transformation	with cartesian transformation #CS ON	with kinematic transformation #TRAFO ON
V.A.MESS.<name>	ACS- Position	transformation of measurement position in the active coordinate system.	transformation of the measurement position in the active coordinate system.
V.A.MOFFS.<name>	= V.A.MESS.<name> - V.A.PROG.<name> - Zero Shifts - Tool Offset	= V.A.MESS.<name> - V.A.PROG.<name> - Zero Shifts - Tool Offset	= V.A.MESS. - V.A.PROG.<name> - Zero Shifts

八、子程序功能

概述

子程序功能是倍福系统的一个特色，在倍福系统中可以实现大量的自定义子程序。这些子程序丰富了 NC 代码功能，并使 NC 代码显得简单易读，功能强大，灵活拓展。CNC 代码中相同的功能或者有一定连续性的 NC 代码使用子程序更方便。倍福 CNC 的子程序包含本地子程序、全局子程序、和 G 代码子程序以及特殊子程序。这些子程序中可以实现变量传递、函数调用等功能。

1、初始子程序

子程序定义：

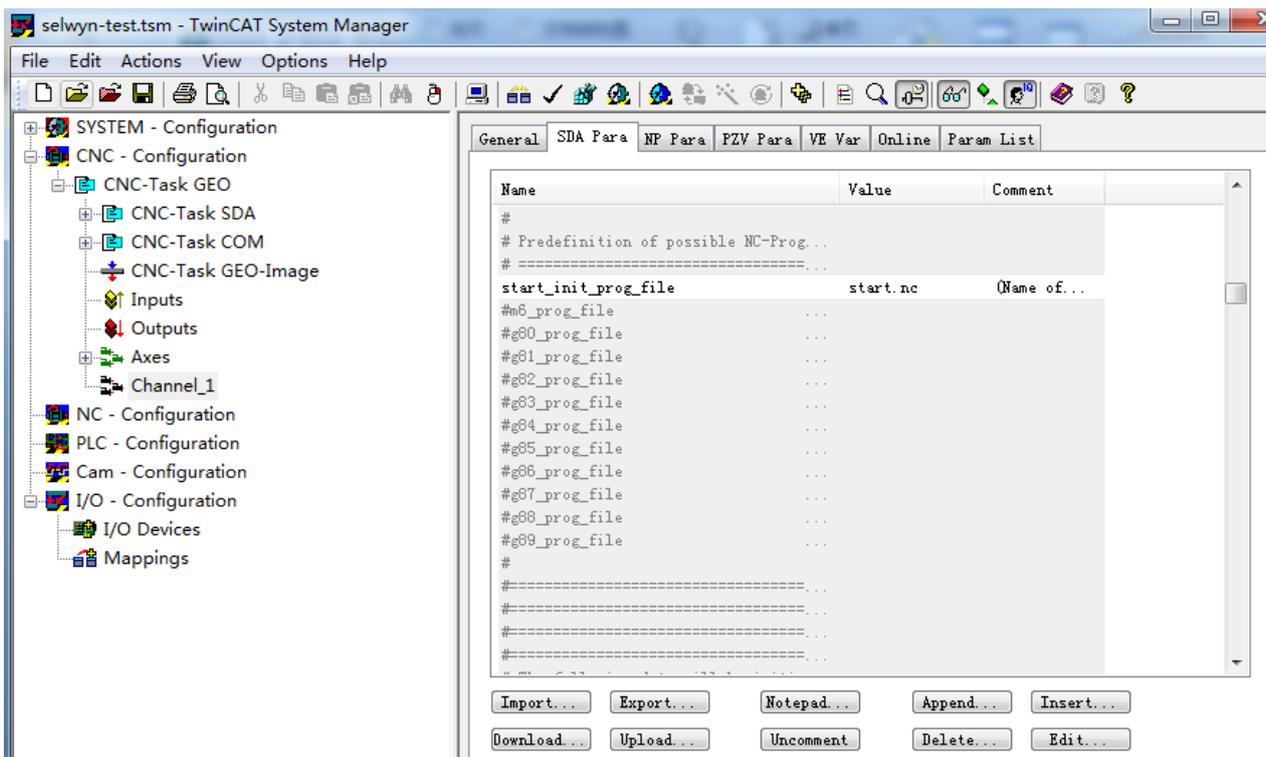
在通道参数中定义初始子程序，start_init_prog_file start.nc（如下图）

子程序说明：

初始子程序应用于自动模式下，当通道参数中设定有初始子程序项时，执行 NC 代码时，将首先执行初始程序中设定的子程序，然后再执行调用的 NC 代码。

子程序作用：

初始子程序通常用于定义机床代码中的初始化，定义或设置坐标系偏置、G 代码预处理功能以及参数定义设定等，方便机床代码的执行。

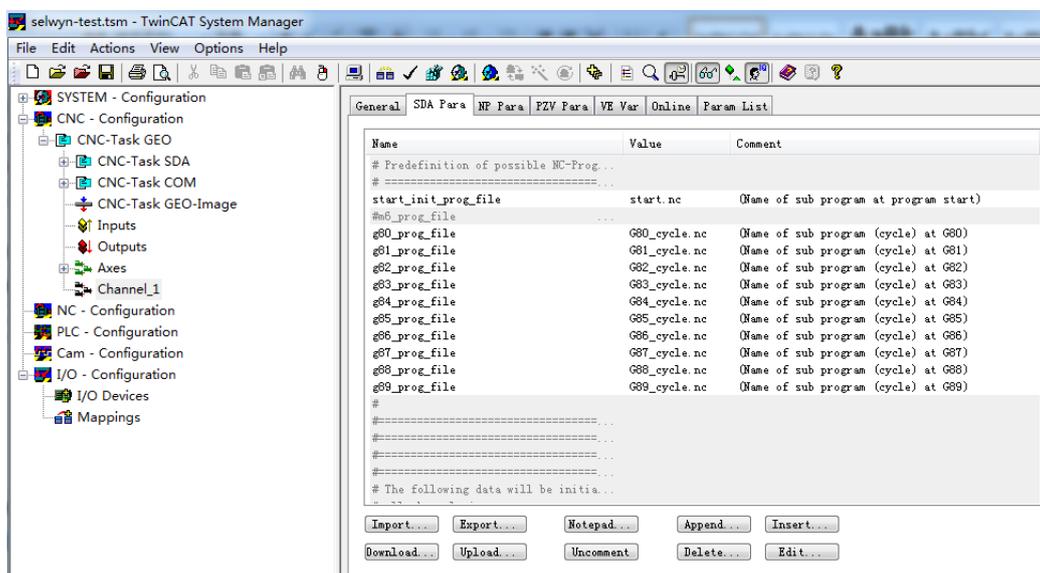


2、G 代码子程序

子程序定义：

G 代码子程序可以通过两种方式定义：

- a) 在通道参数中定义初始子程序，g80_prog_file G80_cycle.nc（如下图）



- b) 通过#FILENAME 方式制定 G 代码；

#FILENAME[G80="g80_up_test.nc"]

c) 两种方式的区别在于，使用通道参数设定的子程序在启动配置文件时即设定完成，不许额外声明，使用#filename 方式设置的子程序需要单独调用声明才能够生效；

子程序说明：

执行代码时，在已经定义完成子程序时，可以单独调用子程序；G 代码子程序可以实现参数传递，即根据需要设定不同的参考变量，影响子程序的执行；例如 G80[100, 200, 300];

G 代码子程序设定范围为 G80-G89\G800-G819 共 30 组。

子程序举例：

G80_cycle.nc

G01 X@P1 Y@P2 Z@P3 F@P4

M60

G02 I10

M17

主程序调用：

N100 #FILENAME[G80="G80_CYCLE.NC"]

N200 G01 X100 Y200 Z300 F1000

N300 G80[200, 300, 400, 2000]

N400 G01 X0 Y0 Z0 F1000

N500 M30

3、本地子程序

本地子程序格式：

LL <string>	(Caution: Blank character between "LL" and <string> is mandatory).
-------------	--

<string> Name of the local sub-routine

本地子程序说明

LL 本地子程序：本地子程序（LUP）

子程序与主程序在同一个文件中，即同一个.NC 文件中；

子程序必须放在主程序之前，并且以%L 开头，空格后位子程序名称如%L L_CYCLE；

子程序必须以%结尾，%结尾之后的第一个程序是主程序；

在主程序中调用子程序格式：LL L_CYCLE,即 LL+空格+名称；

主程序与子程序的变量交互使用 P 变量，提前定义子程序中变量，P1，P2，P3。

本地子程序例程：

```
%L UP1 (第一个本地子程序)
N1 .....
N2 .....
N9 M17 (M17 也可以省略)
%L UP2 (第二个本地子程序)
N11 .....
N12 .....
N19 M29 (M29 也可以省略)
%100 (主程序)
N100 .....
N105 .....
N200 LL UP1 (调用第一个本地子程序子程序)
N250 LL UP2 (调用第二个本地子程序)
N300 M30 ( 主程序结束)
```

4、全局子程序：

全局子程序指令格式：

`L<string> or L <string>`

`<string>`

Name of global sub-routine

全局子程序指令说明：

全局子程序：全局子程序(GUP)

全局子程序与主程序是两个独立的.NC 文件；

全局子程序也包含有局部子程序或者主程序；

主程序调用子程序时，以 L+空格+文件名+.NC，即 L L_CYCLE.NC；

子程序开头不需要以%名称标示；

全局子程序可以在调用主程序的任一位置调用；

可以提前定义子程序中的变量，P1，P2，P3。

全局子程序指令例程：

%L LUP (本地子程序)

N11

N12

N19 M17 (M17 可以被省略)

%333 (主程序)

N100

N110 LL LUP (调用本地子程序)

N200 L GUP_FILE (通过文件名调用全局子程序，在文件中包含着程序的内容)

N300 M30

5、变量调用程序

指令格式:

调用本地子程序:

LL V.E. ... ("LL" 与 V.E. ... 之间必须要有空格).

V.E. ... 通过外部变量来定义子程序名称

调用全局子程序

LV.E. ... or L V.E. ...

V.E. ... 通过外部变量来定义子程序名称

指令说明:

通过 VE 变量代替本地子程序和全局子程序名称。

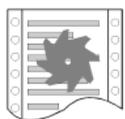
变量名称定义为 string 或 string 数组类型;

LL V.E: V.E 变量指代本地子程序;

L V.E&&LV.E: V.E 变量指代全局子程序。

指令例程

通过外部变量类型为 string 来调用子程序



Programming example 1

Call of sup programs via external variables of type "string"

```
%L TASCHE          (Local sub-routine)
N10 .....
.
N99 M17

%MAIN              (Main program)
N100 .....
N105 .....
N110 LL V.E.LUP    (Call of local sub-routine via external variable)
                   (V.E.LUP, including the string TASCHE)
.
N200 L V.E.GUP     (Call of global sub-routine via external variable)
                   (V.E.GUP, including the string of a file name)
N300 M30
```

通过外部变量为 **STRING** 数组的方式调用子程序



Programming example 2

Call of sup programs via external variables of type "string array"

```

%L TASCHE_1          (Local sub-routine 1)
N10 .....
.
N99 M17

%L TASCHE_2          (Local sub-routine 2)
N10 .....
.
N99 M17

%L TASCHE_3          (Local sub-routine 3)
N10 .....
.
N99 M17

%MAIN                (Main program)
N100 .....
N105 $FOR P1 = 1,3,1
N110 LL V.E.LUP[P1] (Call of sub-routines via external variables)
                        (V.E.LUP[1], V.E.LUP[2],V.E.LUP[3] including)
                        (the strings TASCHE_1, TASCHE_2, TASCHE_3)

N120 $ENDFOR
.

N205 $FOR P2 = 1,5,1
N210 L V.E.GUP[P2] (Call of sub-routines via external variables)
                        (V.E.GUP[1], V.E.GUP[2]..., including the)
                        (strings of the file names)

N220 $ENDFOR

N300 M30

```

6、循环子程序

概述:

循环子程序通常以子程序的方式用于某些特殊机型,例如深孔钻、铣削等机床。这些机床按照设定的格式、参数执行加工动作。

指令格式

L CYCLE [NAME=<cycle_name> [MODAL] @P1=<expr> @P30=<expr>]

NAME=<cycle_name> Name of cycle (file name)

MODAL Modal cycle call. After every further NC block in the main program that contains movement commands, the cycle is implicitly executed again.

The modal effect is cancelled by the NC command #DISABLE MODAL CYCLE.

@P1<expr>... List of the transfer parameters.

... @P30<expr> A maximum of 30 @Pxx parameters of the REAL type can be passed on. Write and read access is only allowed within a cycle. The @Pxx parameters can be assigned direct values, any variables, P parameters and mathematical expressions.

指令说明:

在循环程序中可以使用设定的@P 参数传递设定值;

循环调用必须在一个独立的程序中,不需要更多的 NC 命令,这个语法包含一个设定参数的全局子程序。

循环调用子程序的参数设定中,使用的@P 参数必须明白在子程序中的作用,这个@P 参数可以不赋值,不赋值时,默认为 0; 如果不适用 P 参数的话,可以忽略;

传递参数只能用于相应的子程序,不能用于程序本身;

模态指令取消

#DISABLE MODAL CYCLE

指令举例

以下以钻床举例说明,各种参数设置需要在调用的循环中已经存在与程序文件中(drill.cyc)

钻床循环包含以下参数:

@P1 -> Position of the return plane (absolute)

@P2 -> Position of the machining plane (absolute)

@P3 -> Safety clearance (unsigned)

@P4 -> Final drilling depth (absolute) or

@P5 -> Final drilling depth relative to the machining plane (unsigned)

a) 循环调用使用固定值:

Nxx L CYCLE [NAME=drill.cyc @P1=110 @P2=100 @P3=4 @P4=40]

..

或者设置一个相对钻孔深度 @P5:

..

Nxx L CYCLE [NAME=drill.cyc @P1=110 @P2=100 @P3=4 @P5=60]

..

b) 循环调用使用变量

设定变量必须已经定义再循环程序中

```
#VAR
  V.L.RPL = 110
  V.L.WPL = 100
  V.L.SDST = 4
  V.L.DEP = 50
#ENDVAR
Nxx L CYCLE [NAME=drill.cyc @P1= V.L.RPL @P2=V.L.WPL @P3=V.L.SDST@P4=V.P.DEP]
..
```

c) 循环调用使用 P 变量

这个参数必须提前定义在循环程序之前

```
..
Nxx P10 = 110
Nxx P11 = 100
Nxx P15 = 4
Nxx P17 = 50
Nxx L CYCLE [NAME=drill.cyc @P1= P10 @P2=P11 @P3=P15 @P4=P17]
..
```

d) 循环调用使用数学表达式

```
..
Nxx P20 = 100
Nxx L CYCLE [NAME=drill.cyc @P1= 10+P20 @P2=2*50 @P3=5-1 @P4=P20/2]
..
```

d) 循环调用使用固定值，参数的顺序可以随意设置

```
..
Nxx L CYCLE [@P4=40 NAME=drill.cyc @P2=100 @P3=4 @P1=110]
..
```

e) 循环调用使用固定值，模态指令影响

```
..
Nxx L CYCLE [NAME=drill.cyc @P1=110 @P2=100 @P3=4 @P4=40 MODAL]
..
%drill_main
N05 T1 D1
N10 M06
N15 X0 Y0 Z0 G0 G90 F200 S300 M3 G53 G17
N20 Z110
N30 X40 Y40 (drilling 1)
N40 L CYCLE [NAME=drill.cyc MODAL @P1=110 @P2=100 @P3=2 @P4=55]
N50 X60 Y60 (设定值为模态，在钻孔位置2处执行循环调用)
N60 X100 Y60 (设定值为模态，在钻孔位置3处执行循环调用)
N70 X100 Y20 (设定值为模态，在钻孔位置4处执行循环调用)
#DISABLE MODAL CYCLE
N80 X0 Y0 M5
N100 M30
```

指令应用

Cycle parameters	Description
@P1	Return plane (absolute)
@P2	Reference plane (absolute)
@P3	Safety clearance (relative to the reference plane, unsigned)
@P4	Final drilling depth (absolute)
@P5	Final drilling depth (relative to the reference plane, unsigned)
@P6	Dwell time at final drilling depth
@P10	Machining direction (for offsetting the safety clearance)

Syntax of a cycle call for a through/centring hole:

```
L CYCLE [NAME=drilling.cyc @P1=.. @P2=.. @P3=.. @P4=.. | @P5=.. @P6=.. @P10=..]
```

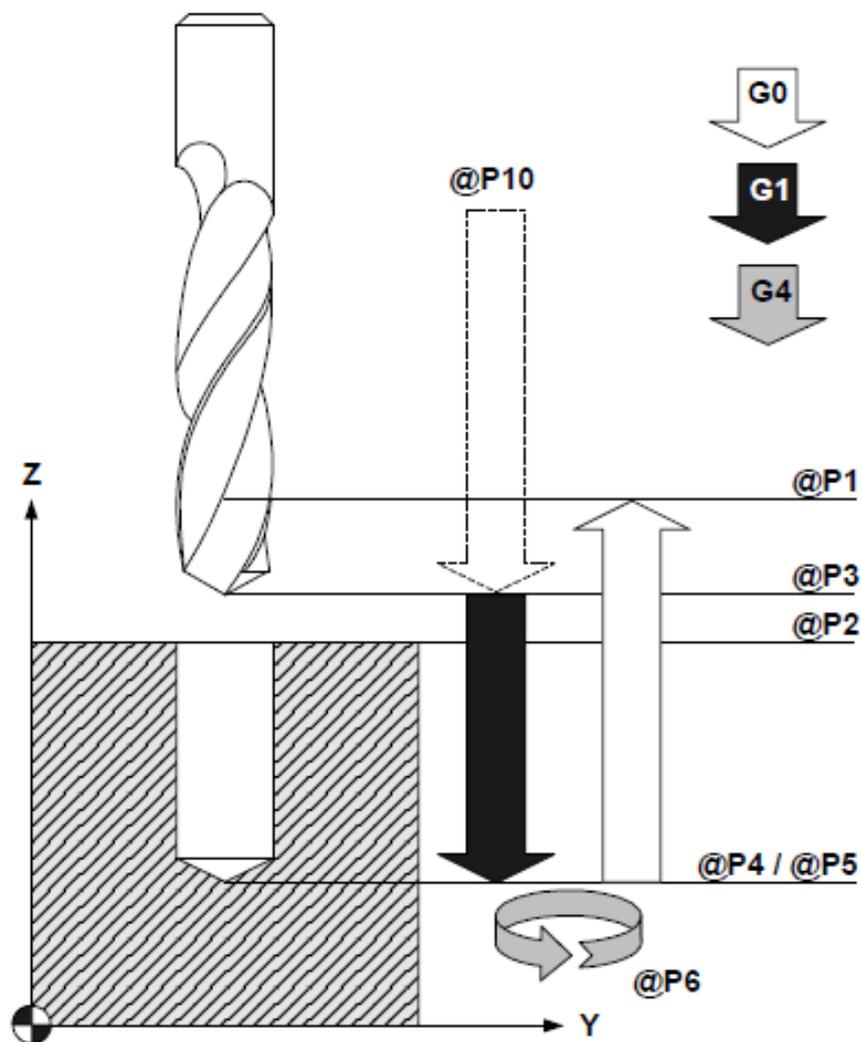


Figure 1-1: Through/centring hole sequence



Programming example

G17
Richtung 30

Rückzugsebene

Sicherheitsabstand

Bezugsebene

Z

Y

20 60

80 5 100

40

```

(Creation of 2 centring holes)
N10 T5 D5           Current tool data
N20 M6             Tool change
N30 G00 G17 G90 F250 M03 S400
                    Technology values
N40 Z100          Travel to return plane
N50 Y20 X0        1st drilling position
N60 L CYCLE [NAME=drilling.cyc @P1=100
@P2=80
                    @P3=5
@P4=40 @P6=1 @P10=30]
N70 Y60 X0        2nd drilling position
N80 L CYCLE [NAME=drilling.cyc @P1=100
@P2=80
                    @P3=5
@P4=40 @P6=1 @P10=30]
N90 Z200 M5       Parking position, spindle
stop
N100 M30          Program end

```

九、变量

概述:

变量以及变量计算, 关于变量部分, 一部分是 CNC 系统内部已经设定名称的参数, 另一部分是自由定义的选择使用的参数。除此之外, NC 通道中设定的外部变量(VE...)

常用语法:

```
V.<NAME_1>.<NAME_2>.<NAME_3>.{<NAME_n>}
```

V. indicates the accessing of the variables

<NAME_1>. the global data designation:

"A." represents axis-specific variables,

"SPDL." represents spindle-specific variables,

"G." represents inter-axis, globally valid variables,

"E." represents external variables,

"P." represents self-defined, non-inter-program, non-global variables,

"L." represents self-defined, non-inter-program, local variables,

"S." represents self-defined, inter-program, global variables.

<NAME_2>. specifies the data name

<NAME_3>. e.g. the index, if several data of the same kind are to be distinguishable.

轴号定义:

插补轴:

变量的最后一位代码表示设定轴号和群主设定变量, 所以在设定参数的时候可以设定如下方式:

".X" or "[0]",

".Y" or "[1]",

".Z" or "[2]"

当通道中参数 `gruppe[j].achse[i].*`, 'X' 轴对应的 index 序号为 0; 'Y' 轴对应的 index 序号为 1; 'Z' 轴对应的 index 序号为 2;

举例:

X 轴的绝对坐标值:

V.A.ABS.X OR V.A.ABS[0]

主轴:

主轴设定与插补轴类似，可以使用主轴名称或者相应的主轴数值，如下：

".S" or "[0]",
".S2" or "[1]", ...

举例:

V.SPDL.LOG_AX_NR.S OR V.SPDL.LOG_AX_NR[0]

变量应用举例:

V.A 变量使用

本例列举为 X 轴方向上的直线插补，移动距离均为坐标偏置后的 Y 轴位置

N10 G92 X0 Y40 Z0 (Position preset)

N20 G91 G01 F1000 XV.A.BZPY

N30 XV.A.BZP[1] (here: Axis index 1 == Y)

N40 M30

V.G 变量使用:

坐标偏置表格，通过 VG 变量对坐标偏置进行赋值，此值为第二个坐标偏置表格的第一个 index 数值赋值。

N10 V.G.NP[2].V[1] = 100

...

EXIST 功能使用

通过 EXIST 功能检测 VG 变量如当前例程轴号，判断执行特殊轴操作。

N10 G90 Y0

N20 \$IF EXIST[V.A.LOG_AX_NR.X] == TRUE

N30 X-10 (X-axis is in channel, move to position -10)

N40 \$ELSE

N50 #CALL AX [X,1,0] (X-axis is not in channel, first request axis)

N60 \$ENDIF

...

M30

1、VA 变量

V.A.变量只用于直线、旋转轴的参数编程，不能用于主轴，主轴单独设定。

V. A. <var_name>	Significance	Data type	Unit of In/ Output	R / W
MENT. X	Mental coordinate of the previous NC-block (see chapter Mirroring G20-G23)	Real	[mm] , [inch]	R
PROG. X	Programmed coordinate of the previous NC block During active contour rotation (#ROTATION) the variable delivers the coordinate value mapped on the machine axes.	Real	[mm] , [inch]	R
ABS. X	Absolute coordinate of the previous NC block respectively currently absolute coordinate after NC command #CHANNEL_INIT in the currently active coordinate system each.	Real	[mm] , [inch]	R
ACT_POS. X	Current actual position in the present coordinate system without any offsets	Real	[mm] , [inch]	R
-SWE. X	Current effective negative software limit switch	Real	[mm] , [inch]	R
+SWE. X	Current effective positive software limit switch	Real	[mm] , [inch]	R
-SWE_MDS. X	Configured negative software limit switch (acc. to machine parameter [2]-10)	Real	[mm] , [inch]	R
+SWE_MDS. X	Configured positive software limit switch (acc. to machine parameter [2]-11)	Real	[mm] , [inch]	R
REF. X	Machine reference point (is documented only after successful machine reference search)	Real	[mm] , [inch]	R
BZP. X	Coordinate preset	Real	[mm] , [inch]	R
PZV. X	Machine table offset	Real	[mm] , [inch]	R
MESS. X	After successful measuring traverse the variable (measured value) gives the axis-specific coordinate in which all re-locations are included (see programming example).	Real	[mm] , [inch]	R
MOFFS. X	Measurement offset	Real	[mm] , [inch]	R
MERF. X	Measurement traverse completed? If yes, then 1	Boolean	0 , 1	R
MEIN. X	Included measurement offset	Real	[mm] , [inch]	R
RERF. X	Machine reference search completed ? If yes, then 1	Boolean	0 , 1	R
MANUAL_OFFSETS. X or SOFFS. X	Moving path during manual operation. Useful only combined with NC command #GET MANUAL OFFSETS	Real	[mm] , [inch]	R
MODE. X	Current axis mode	Integer	-	R
MODULO_VALUE. X	Modulo range	Real	[°]	R
LOG_AX_NR. X	Logical axis number of an axis	Integer	-	R
AX_LIST_NAME. X	Configured axis name (acc. to machine parameter [2]-18)	String	-	R
MIRROR. X	Mirror mode of the axis: (1: no mirroring -1: axis is mirrored)	Integer	-	R
WCS. X MCS. X	Conversion between machine coordinates (MCS) and work piece coordinates (WCS). Only useful in combination with the NC commands #WCS TO MCS and #MCS TO WCS	Real	[mm] , [inch]	R / W

V.A 变量应用举例

例程注释:

设定工件坐标系 X50 位置为原点，工件坐标系内，执行 X 轴移至 X100（机械坐标系 X150）位置过程中，检测外部信号，判断到位。遇到传感器信号时，停止执行 N20 动作，将坐标系还原，使用测量程序的位置，执行 XY 轴位置移动。

N10 G90 G92 X50

N20 G100 X100 (Measurement traverse, measurement interrupt)

(occurs 2mm before the target)

N30 G90 G92 X0

N40 XV.A.MESS.X YV.A.MOFFS.X

2、V. SPDL

主轴设定:

主轴功能通过通道参数读取。

V.SPDL.<var_name>	Significance	Data type	Unit of In/ Output	Enabled access Read /Write
LOG_AX_NR.S	Logical axis number of a spindle	Integer	-	R
PLC_CONTROL.S	Is spindle a PLC spindle? If yes, then 1	Boolean	0 , 1	R
NBR_IN_CHANNEL	Total number of available spindles in the current NC channel	Integer	-	R
M_FCT_FREE	How is the classification of the M functions M3,M4, M5, M19? Explicitly defined as spindle M functions: 0 Free available for other techno functions: 1 Caution: Write access causes permanent actualisation of internal channel parameter data (P-CHAN-00098)			

主轴功能参数引用举例:

通过 EXIST 功能判断通道中，主轴是否存在。

```
N10 G90 Y0
```

```
N20 $IF EXIST[V.SPDL.LOG_AX_NR.S] == TRUE
```

```
N30 M3 S1000 (Spindle S with speed 1000 rpm)
```

```
N40 $ELSE
```

```
N50 #MSG ["Spindle S is not available!"] (Message output and stop)
```

```
N55 M0
```

```
N60 $ENDIF
```

```
...
```

```
M30
```

3、V.G 全局变量

概述:

V.G 变量用于读写内部轴参数，全局通道参数。

读取轴设定代码，下表参数为用于轴参数读取，例如刀具补偿中的 X 轴偏置参数设定、读取等；

示例如下：

V.G.WZ[j].V.X or V.G.WZ[j].V[0]

1、V.G.<var_name>变量:

V. G. <var_name>	Significance	Data type	Unit of In/ Output	R /W
BLOCK_NR	The last programmed NC-block number	Integer	-	R
PROG_SA	Local block type, if in NC-block no more alteration takes place	Integer	-	R
MASS_MM	If measuring unit [mm], then 0	Boolean	0 , 1	R
MASS_360	If measuring unit [degree], then 0	Boolean	0 , 1	R
I	I-coordinate of the circular programming. Under active center point correction (G165), access on corrected value	Real	[mm] , [inch]	R
J	J-coordinate of the circular programming. Under active center point correction (G165), access on corrected value	Real	[mm] , [inch]	R
K	K-coordinate of the circular programming. Under active center point correction (G165) , access on corrected value	Real	[mm] , [inch]	R
R	The radius traversed during circular interpolation	Real	[mm] , [inch]	R
FEEDRATE	The last programmed feedrate (F word)	Real	[mm/min], [m/min], [inch/min]	R
MERR[i]	Correction offset of the circle center point	Real	[mm] , [inch]	R
TIMER[i]	Counter value of the timer with number <i>	Integer	[ms]	R

RANDOM	Creates a random value in the range 0.0 – 1.0	Real	–	R
PROG_ABS	Dimensioning, 0: G91 active 1: G90 active	Boolean	0 , 1	R
KIN[j].PARAM[i]	Kinematic parameters of a specific kinematic [j] Caution: Write access causes permanent actualisation of internal channel parameter data! Value must be programmed in internal units!	Real	[0.1 µm], [10–4 deg]	R / W
CNC_RELEASE	Build number of CNC version	Integer	–	L

MAIN_FILE_NAME	File name of the NC main program	String	–	L
MAIN_PROG_NAME	Name (%...) of the NC main program	String	–	L
MAIN_PROG_NR	Number of the NC main program, if the program name is a digit	Integer	–	L
FILE_NAME	File name of the currently active NC program	String	–	R
PROG_NAME	Name (%...) of the currently active NC program	String	–	R
PROG_NR	Number of the currently active NC program, if the NC-program name is a digit	Integer	–	R
PATH_NR	Logical path number of the currently active NC main or global sub program according to the start-up list (P-STUP-00019)	Integer	–	R
FILE_OFFSET	File position of the beginning of the NC block with the programmed V.G. FILE_OFFSET variable	Integer	–	R

2、V.G.WZ[j]变量

WZ[j]变量用于读取任意刀具的数据信息。这个变量既可以读取外部刀具数据，也可以读取使用的内部刀具数据 ‘j’ 对应于刀具号与刀具列表中的 index 号相对应。

V. G. <var_name>	Significance	Data type	Unit of In/ Output	Read /Write
WZ[j].R	Radius of the tool	Real	[mm] [inch]	R
WZ[j].L	Length of the tool	Real	[mm] [inch]	R
WZ[j].P[i]	Parameters of the tool	Real	-	R
WZ[j].V[i]	Axis-offsets of the tool	Real	[mm] [inch]	R
WZ[j].ME	Measuring unit of radius, length and axes offsets, provides always 0 (for [mm]) when using a tool list, otherwise the variable has no significance	Boolean	0 , 1	R
WZ[j].OK	Valid flag of the tool; if valid, then 1	Boolean	0 , 1	R
WZ[j].SPDL_AX_NR	Logical axis number of the assigned spindle	Integer	-	R
WZ[j].KIN_PARAM[i]	Kinematic parameters of the tool	Real	[0.1 μm], [10 ⁻⁴ deg]	R
WZ[j].KIN_ID	Kinematik-ID of the tool	Integer	-	R
WZ[j].TYPE	Tool type (0: Milling 1: Turning 2: Grinding)	Integer	-	R
WZ[j].SRK_ID	Cutter orientation of a turning tool	Integer	-	R
WZ[j].S_MIN_SPEED	Minimum speed (Tool dynamic data)	Real	[U/min]	R
WZ[j].S_MAX_SPEED	Maximum speed (Tool dynamic data)	Real	[U/min]	R
WZ[j].S_MAX_ACC	Maximum acceleration (Tool dynamic data)	Real	[deg/sec ²]	R
WZ[j].SISTER_VALID	Valid flag of the sister tool (TOOL-ID)	Boolean	0 , 1	R
WZ[j].SISTER	Number of the valid sister tool	Integer	-	R
WZ[j].VARIANT_VALID	Valid flag of the variant tool (TOOL-ID)	Boolean	0 , 1	R
WZ[j].VARIANT	Number of the valid variant tool	Integer	-	R

3、V.G. WZ_AKT /T_AKT/ D_AKT 变量

概述:

‘WZ_AKT’ 变量、‘T_AKT’、‘D_AKT’ 用于访问当前选用的刀具信息。这些参数都可以用于外部刀具管理以及内部刀具管理的参数读写。

选择新的刀具时，需要重新写入当前选择的刀具信息。当取消选择 D0 或者选用一个新的刀具 D**时，这些数据发生了改变。不会继续保存。

例外的情况是，如果使用外部刀具表格，在取消选择刀具或者选额新的刀具时，刀具信息（V.G.WZ_AKT.P[i]）通过 d_clear_to_wzv 参数设定仍然保存在设定的配置表格中。

V.G.T_AKT/ D_AKT

T_AKT	Number of the selected tool	Integer	-	R
D_AKT	Number of the selected tool compensation data set	Integer	-	R

V.G. WZ_AKT

V. G. <var_name>	Significance	Data type	Unit of In/ Output	Read /Write
WZ_AKT.R	Radius of the current tool	Real	[mm] [inch]	R / W
WZ_AKT.L	Length of the current tool	Real	[mm] [inch]	R / W
WZ_AKT.P[i]	Parameters of current tool	Real	-	R / W
WZ_AKT.V[i]	Axis offset of the current tool	Real	[mm] [inch]	R / W
WZ_AKT.ME	Measuring unit of radius, length and axes offsets of the current tool, provides always 0 (for [mm]) when using a tool list, otherwise the variable has no significance	Boolean	0 , 1	R
WZ_AKT.OK	Tool valid flag of the current tool; if valid, then 1	Boolean	0 , 1	R
WZ_AKT.SPDL_AX_NR	Logical axis number of the assigned spindle	Integer	-	R
WZ_AKT.KIN_PARAM[i]	Kinematic parameter of the current tool. Note: For write access value must be programmed in internal units!	Real	[0.1 μm], [10-4 deg]	R / W

WZ_AKT.KIN_ID	Kinematic-ID of the current tool	Integer	-	R
WZ_AKT.TYPE	Tool type of the selected tool (0: Milling 1: Turning 2: Grinding)	Integer	-	R
WZ_AKT.SRK_ID	Cutter orientation of the selected turning tool	Integer	-	R
WZ_AKT. S_MIN_SPEED	Minimum speed (Tool dynamic data)	Real	[rpm]	R
WZ_AKT. S_MAX_SPEED	Maximum speed (Tool dynamic data)	Real	[rpm]	R
WZ_AKT. S_MAX_ACC	Maximum acceleration (Tool dynamic data)	Real	[deg/sec ²]	R
WZ_AKT. SISTER_VALID	Valid flag of the sister tool (TOOL-ID)	Boolean	0 , 1	R
WZ_AKT.SISTER	Number of the valid sister tool	Integer	-	R
WZ_AKT. VARIANT_VALID	Valid flag of the variant tool (TOOL-ID)	Boolean	0 , 1	R
WZ_AKT.VARIANT	Number of the valid variant tool	Integer	-	R
WZ_AKT. WEAR_RADIUS	Radius wear at radius compensation (OTC)	Real	[mm] , [inch]	R
WZ_AKT.WEAR[i]	Wear in axis <i> at length compensation (OTC)	Real	[mm] , [inch]	R
WZ_AKT. WEAR_CONST	Wear constant (OTC)	Real	[0.1 μm/m]	R / W

WZ_INFO[i]参数

WZ_INFO[i]参数可以取代 V. G. WZ_AKT.P[i]参数访问外部刀具参数。这个参数只能用于外部刀具管理表。

V. G. <var_name>	Significance	Data type	Unit of In/ Output	Read /Write
WZ_INFO[i]	Parameters of current tool	Real	-	R

V.G.WZR/WZL 参数

V.G.WZR/WZL 参数用于刀具补偿。

V. G. <var_name>	Significance	Data type	Unit of In/ Output	Read /Write
WZR	Radius of the current tool (Old syntax)	Real	[mm] [inch]	R / W
WZL	Length of the current tool (Old syntax)	Real	[mm] [inch]	R / W

V.G.<NP[J]>坐标偏置

V.G.NP[j].V[i]刀具偏置读取设定;

V. G. <var_name>	Significance	Data type	Unit of In/ Output	Read /Write
NP[j].V[i]	Zero offset of one axis. Caution: Write access causes permanent actualisation of internal zero offset data!	Real	[mm] [inch]	R / W
NP[j].ALL	Addressing of all axes of a zero group. Caution: Write access causes permanent actualisation of internal zero offset data!	Real	[mm] [inch]	R / W
NP_AKT.V[i]	Currently active zero offset of an axis	Real	[mm] [inch]	R / W
NP_AKT.ALL	Addressing of the currently active zero offsets of all axes	Real	[mm] [inch]	R / W
NP_AKT.IDX	Index of the currently active zero offset group (e. g. 0 at G53, 1 at G54...)	Integer	-	R
NP_DEFAULT	Default index of zero offset group after start-up	Integer	-	R / W

V.G.<CS/ACS>坐标偏置

V.G.CS/ACS 读取坐标偏置信息

V. G. <var_name>	Significance	Data type	Unit of In/ Output	Read /Write
CS_ACTIVE	CS (#CS···) active? If active, then 1	Boolean	0 , 1	R
CS_COUNT	Number of active (linked) CS	Integer	-	R
ACS_ACTIVE	ACS (#ACS···) active? If active, then 1	Boolean	0 , 1	R
ACS_COUNT	Number of active (linked) ACS	Integer	-	R
CS	Machining coordinate system (CS) on/off	Integer	-	R
ACS	Fixture adaptive coordinate system (ACS) on/off	Integer	-	R
AKT_PLATZ	Current clamp position offset index			R
CNC_CHANNEL	Current channel number	Integer	-	R
IPO_COUNT	System time counter	Integer	-	R
TOOL_COMP	Active tool compensation (1: RTCP 2: TLC 3: 2.5D)			R

V.G.< AX_LINK >耦合功能

V. G. <var_name>	Significance	Data type	Unit of In/ Output	Read /Write
AX_LINK.NR	Number of current or last active coupling group	Integer	-	R
AX_LINK.ACTIVE	Is an axes coupling (#AX LINK ON) active? If active, then 1	Boolean	0 , 1	R
AX_LINK_GROUP[i].ACTIVE	Is a specific coupling group with number [i] active? If active, then 1	Boolean	0 , 1	L

V.G.< ROTATION>旋转坐标系

V. G. <var_name>	Significance	Data type	Unit of In/ Output	Read /Write
ROT_ACTIVE	Contour rotation (#ROTATION...) active? If active, then 1	Boolean	0 , 1	R
ROT_ANGLE	Angle of contour rotation	Real	[deg]	R
ROT_CENTER1	Offset of first main axis to the rotation point at contour rotation	Real	[mm] [inch]	R
ROT_CENTER2	Offset of second main axis to the rotation point at contour rotation	Real	[mm] [inch]	R

V.G.< WCS_POSLIMIT >

主坐标轴的限位读取;

V. G. <var_name>	Significance	Data type	Unit of In/ Output	Read /Write
WCS_POSLIMIT_1	Motion limits in the main axes in WCS. Only useful in combination with the NC command #GET WCS POSLIMIT	Real	[mm]	R
WCS_POSLIMIT_2			[inch]	
WCS_POSLIMIT_3				

V.G.<FUNC_ACTIVE>

CNC 功能使用读取

V. G. <var_name>	Significance	Data type	Unit of In/ Output	Read /Write
MIRROR_ACTIVE	Mirroring (G351 X... oder G21, G22, G23) active? If active, then 1	Boolean	0 , 1	L
BLOCK_SEARCH_ACTIVE	Block search active? If active, then 1	Boolean	0 , 1	R
TRAFO_ACTIVE	Kinematic transformation (#TRAFO...) active? If active, then 1	Boolean	0 , 1	R
MCS_ACTIVE	Temporary transition to the machine axes coordinate system (#MCS...) active? If active, then 1	Boolean	0 , 1	R
HSC_ACTIVE	Free form surface operation (#HSC...) or spline interpolation (#SET SPLINE... oder G151) active? If active, then 1	Boolean	0 , 1	R
CONT_MODE_ACTIVE	Polynom contouring (G261) active? If active, then the variable provides the value of the current contouring type: 1 at contouring type 1 (AUTO_DEV) 2 at contouring type 2 (AUTO_VEL) 3 at contouring type 3 (DIST) 4 at contouring type 4 (DEV) 5 at contouring type 5 (POS) 6 at contouring type 6 (DIST_SOFT) 0 at G260	Integer	-	R

TRC_ACTIVE	Tool radius compensation (G41, G42) active? If active, then 1	Boolean	0 , 1	R
OTC_ACTIVE	Online tool compensation (#OTC...) active? If active, then 1	Boolean	0 , 1	R

V.G.<M_FCT/H_FCT>

V.G.<M_FCT/H_FCT>参数用于读写 M/H 功能数据及通道参数。

如果对同步类型不一致的 M/H 功能函数进行写入数值，同步类型会与之相对应修改。这个改变会一直生效，知道下一次激活。如果读取提前输入值，此时的 M/H 功能不一致的话，会触发报警信息。

V. G. <var_name>	Significance	Data type	Unit of In/ Output	Read /Write
M_FCT[i].SYNCH	Synchronisation type of a M function <i>, defined in the channel parameters [1]-3	Integer	-	R / W
M_FCT[i].PRE_OUTP_PATH	Pre output distance m_pre_output [1]-47 of a M function <i> from synchronisation type MEP_SVS, defined in the channel parameters.	Real	[mm] [inch]	R / W
M_FCT[i].PRE_OUTP_TIME	Pre output time m_pre_output [1]-47 of a M function <i> from synchronisation type MET_SVS, defined in the channel parameters.	Real	[sec]	R / W
H_FCT[i].SYNCH	Synchronisation type of a H function <i>, defined in the channel parameters [1]-3	Integer	-	R / W
H_FCT[i].PRE_OUTP_PATH	Pre output distance h_pre_output [1]-47 of a H function <i> from synchronisation type MEP_SVS, defined in the channel parameters.	Real	[mm] [inch]	R / W
H_FCT[i].PRE_OUTP_TIME	Pre output time h_pre_output [1]-47 of a H function <i> from synchronisation type MET_SVS, defined in the channel parameters.	Real	[sec]	R / W

V.G.SPEED_LIMIT 参数

V.G.SPEED_LIMIT 参数用于访问定义在通道的速度限制前馈参数。

当对一个输出到通道的参数进行写入的时候，可能会打断当前的运动执行，如果写入一个参数但是距离参数的单位错误时，单位会适应写入值。写入的参数一直生效，直到程序复位。当程序开始执行的时候，这个通道参数再次生效。如果读取一个数值，但是该数值的距离参数与输入的数值单位不同，会发出报警信息。

SPEED_LIMIT. ENABLE	Selection/deselection of speed limit look ahead (0: Deselection 1: Selection)	Boolean	0 , 1	R / W
SPEED_LIMIT. VEL_LIMIT	Definition of speed limit value	Integer	[%]	R / W
SPEED_LIMIT. TIME	Unit definition of the distances to and from the "corner" (0: Path 1: Time)	Boolean	0 , 1	R / W
SPEED_LIMIT. DIST_TO_CORNER	Path distance to "corner"	Real	[mm] , [inch]	R / W
SPEED_LIMIT. DIST_FROM_CORNER	Path distance from "corner"	Real	[mm] , [inch]	R / W
SPEED_LIMIT. TIME_TO_CORNER	Time distance to "corner"	Real	[sec]	R / W
SPEED_LIMIT. TIME_FROM_CORNER	Time distance from "corner"	Real	[sec]	R / W
SPEED_LIMIT. OVERRIDE_WEIGHT	Weighting of speed limit value(0: no weighting 1: Weighting by override)	Boolean	0 , 1	R / W

V.G.<SUB_PROG>子程序相关

子程序循环是否执行，以及子程序中的传递函数是否执行。

CYCLE_ACTIVE	Is the current program level a cycle? If the current program was called with L or LL CYCLE... or G8xx with transfer parameters, then 1	Boolean	0 , 1	R
@P[i].VALID	Is a specific transfer parameter programmed in cycle call or sub programm call G8xx ? 0: Parameter is not programmed 1: Parameter is programmed Index <i> contains the number of the parameter, which should be checked for validity (e.g. 1 for @P1 or the first parameter, called with G8xx). Note for read access: The variable can only be used inside a cycle or in global sub programm G8xx with transfer parameters !	Boolean	0 , 1	R
TIME_STAMP	Current time stamp <date time> : (e.g. 16.06.2011 13:08:10:000) [from V2.10.1507.02]	String	-	R

4、自定义变量

概述:

自定义变量可以在 NC 主程序总定义,也可以在子程序中进行定义。自定义变量定义的方式是将变量名通过#VAR 开头, #ENDVAR 结尾的变量声明。

变量声明语法:

V.P 和 V.S 变量可以在变量声明结构块之外生成,也就是将第一次写入该变量作为变量创建的声明。但是 V.L 变量以及数组的创建必须在#VAR 与#ENDVAR 组成的声明结构中声明。为了更好的区别初始化变量数组,使用“\”字符对数组进行区分。

#VAR	Start of declaration block
:	
:	Declaration and initialisation part
:	
#ENDVAR	End of declaration block

程序实例:

```
%prog_name
```

```
:
```

```
#VAR
```

```
V.P.ARRAY_1[3][6] = [10,11,12,13,14,15, \
                    20,21,22,23,24,25, \
                    30,31,32,33,34,35 ]
```

```
V.L.MY_ARRAY[3][6] = [10,11,12,13,14,15, 20,21,22,23,24,25, 30,31,32,33,34,35]
```

```
V.P.VAR_1
```

```
V.L.VAR_1
```

```
V.S.VAR_1
```

```
#ENDVAR
```

变量删除语法:

自定义变量和数组可以在 NC 程序中进行删除, 采用#DELETE 命令进行删除。语法结构如下:

```
#DELETE V.<name> {, V.<name>}
```

程序实例:

```
#DELETE V.P.ARRAY_1, V.L.MY_ARRAY, V.P.VAR_1, V.L.VAR_1, V.S.VAR_1
```

特殊语法使用:

SIZEOF 和 EXIST 功能可以被用来判断变量数组的容量和检验自定义变量是否存在。

程序实例:

通过 EXIST 指令对一个自定义的 V.S.-数组变量的有效性进行检验, 是否这个变量数组已经在前面部分的程序中已经定义完成, 是否需要进行继续定义。

```
N10 $IF EXIST[V.S.EXAMPLE[0]] == TRUE
```

```
N20 V.S.EXAMPLE[2] = 10 (enter value for V.S-variable[2])
```

```
N30 $ELSE
```

```
N40 #VAR
```

```
N50 V.S.EXAMPLE[5] = [1,2,3,4,5 ]
```

```
N60 #ENDVAR
```

```
N60 $ENDIF
```

```
...
```

```
M30
```

5、V.P.<VAR-NAME>全局变量，执行程序结束失效

概述：

使用 V.P.变量，可以定义单独的变量并且对变量进行赋值。V.P.变量在执行的 NC 程序中全局有效（V.P.变量创建完成后，在当前程序的任意位置以及所有的子程序中都有效），但是当程序执行结束后，变量失效。

V.P.变量语法：

V.P.<FREE_DEF> global variable, not valid after program end

<FREE_DEF> whereby FREE_DEF is a name chosen arbitrarily, which can consist of a maximum firmly given number of any characters (excluding blank characters, tabulators, commentaries, comparison operators, mathematic operators, brackets).

V.P.变量程序示例：

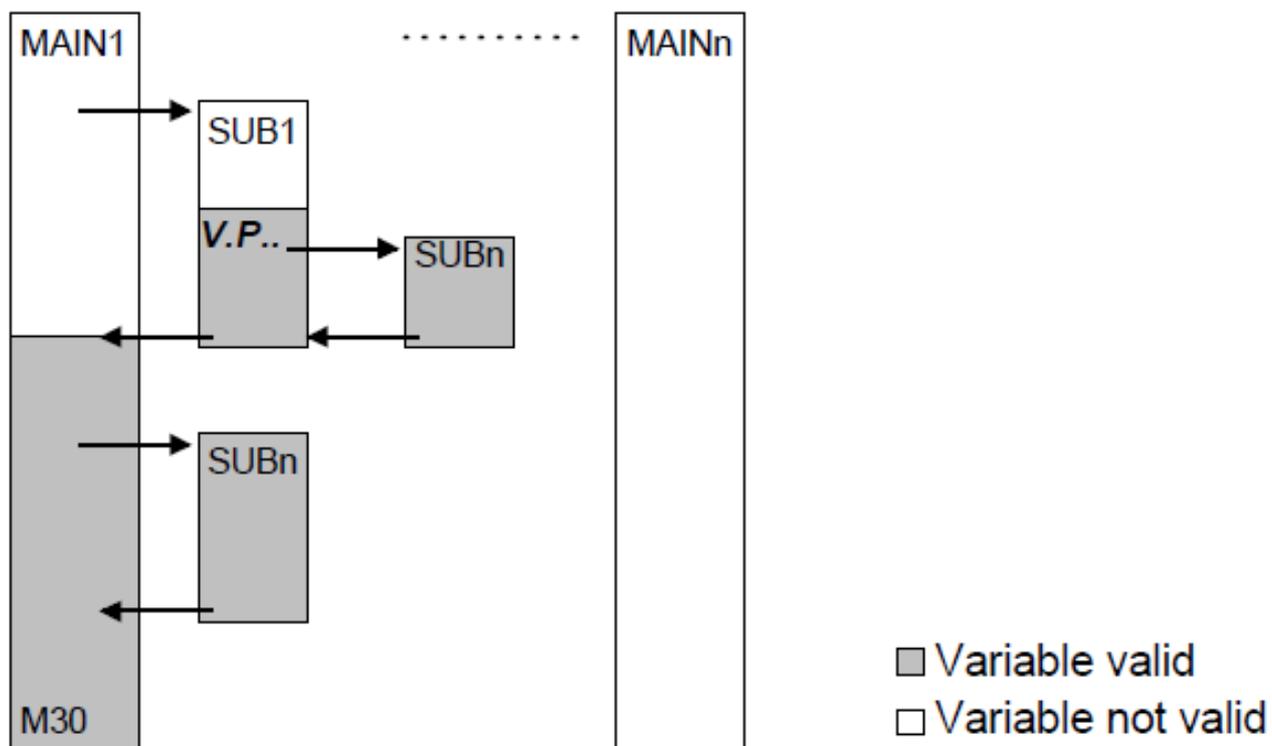
创建一个变量 V.P.VAR_5 并初始化赋值 20。程序中的直线插补指令在 X 方向上，执行指令的数值为 20.

```
#VAR
V.P.VAR_5
#ENDVAR
:
N10 V.P.VAR_5 = 20
N20 XV.P.VAR_5
.....OR.....
#VAR
V.P.VAR_5 = 20
#ENDVAR
:
N20 XV.P.VAR_5
:
```

V.P...变量有效范围:

自定义变量 V.P 参数最大数量在倍福系统时 1000 个，重新启动程序时，所有的 V.P...变量名称以及数值将会被删除。

下图显示的是 V.P...变量的有效范围示意图



6、V.S.<VAR-NAME>全局变量，执行程序结束仍然有效

概述：

V.S...变量，可以单独定义变量并对变量进行赋值。这个变量可以在仍以主程序、子程序有效，并且对后续的主程序相同的变量名称仍然有效。变量数值修改只能通过覆盖写入的方式。这个变量名称可以通过重启系统或者初始化控制的方式删除。

V.S...变量语法：

V. S.<FREE_DEF> global variable, valid after program end

<FREE_DEF> whereby FREE_DEF is a name chosen arbitrarily, which consists of characters (excluding blank characters, tabulators, commentaries, comparison operators, Mathematic operators, brackets).

V.S...示例程序：

创建一个数组变量 V.S.VAR[5]并对其进行初始赋值。之后在 X 方向进行一个直线插补，变量数值为 V.S.VAR[4](X20)。

```
#VAR
```

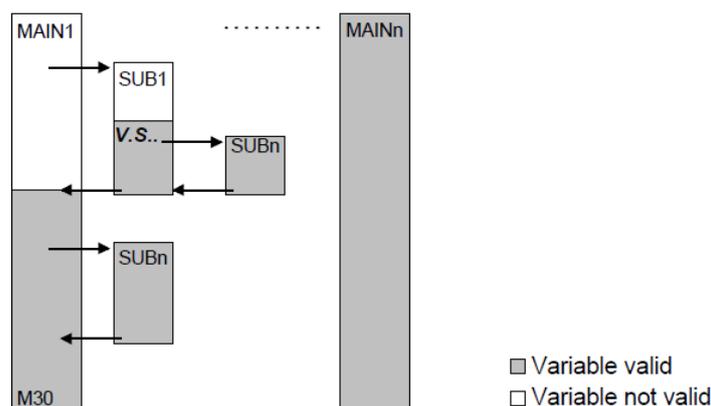
```
V.S.VAR[5] = [5,10,10,15,20]
```

```
#ENDVAR
```

```
N20 XV.S.VAR[4]
```

V.S...变量有效范围：

自定义 V.S...变量最大数量是 400 个。



7、V.L.<VAR-NAME>局部变量，执行程序结束后失效

概述：

自定义变量“V.L...”单独定义变量并对其赋值。这个变量是局部变量，用于当前的子程序和其调用的子程序。执行完当前的子程序后，将被删除。

V.L...变量语法：

V. L.<FREE_DEF> local variable, not valid after program end

<FREE_DEF> whereby FREE_DEF is a name chosen arbitrarily, which consists of characters
(excluding blank characters, tabulators, commentaries, comparison operators,
Mathematic operators, brackets).

V.L...变量程序示例：

创建一个“V.L.LOC_VAR”变量，并对其赋初始值 10,。在之后的程序中调用这个数值，执行在 X 方向上的移动(X10)。

```
#VAR
```

```
V.L.LOC_VAR
```

```
#ENDVAR
```

```
N10 V.L.LOC_VAR = 10
```

```
N20 XV.L.LOC_VAR
```

```
:
```

```
or
```

```
:
```

```
#VAR
```

```
V.L.LOC_VAR = 10
```

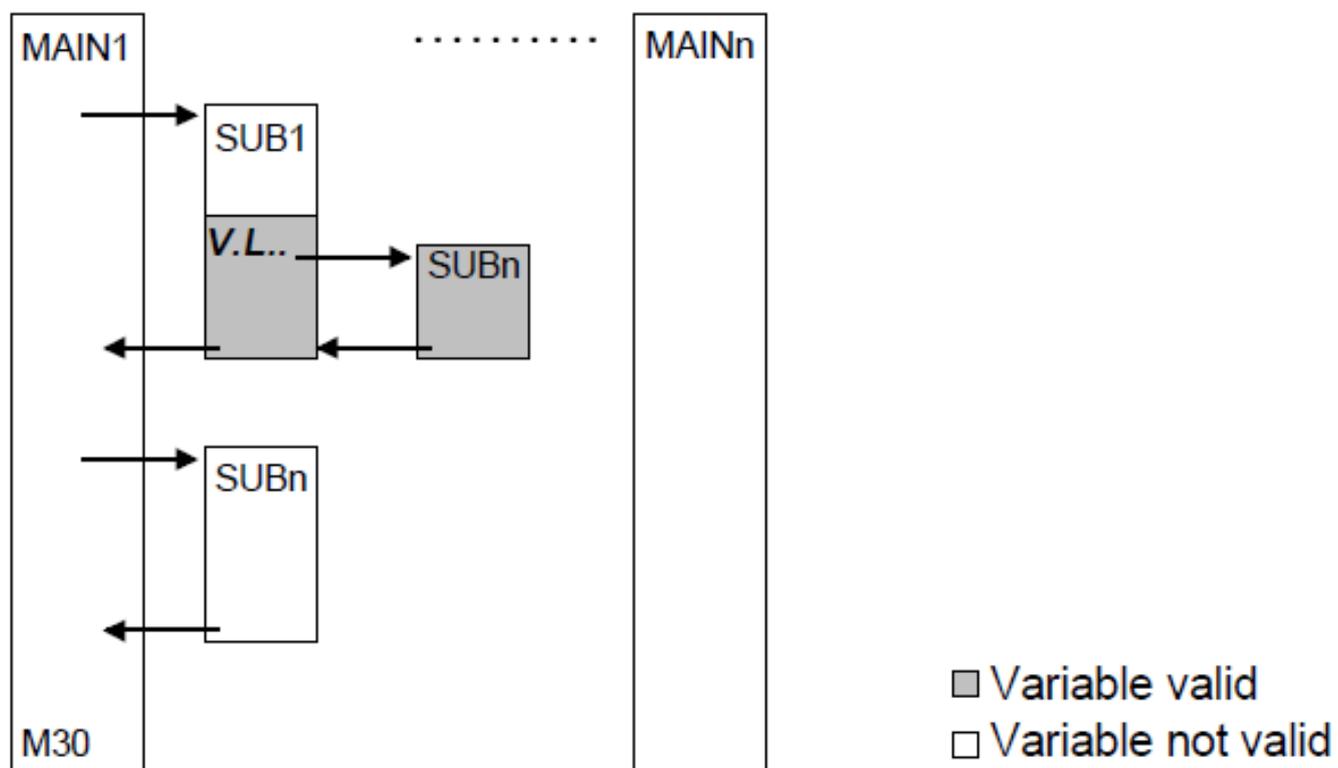
```
#ENDVAR
```

```
N20 XV.L.LOC_VAR
```

V.L...变量的有效范围:

自定义变量 V.L...变量的最大数值为 50 个，当启动新程序时，所有的 V.L...变量以及数值将被删除。

下图指示的是 V.L...变量的有效范围

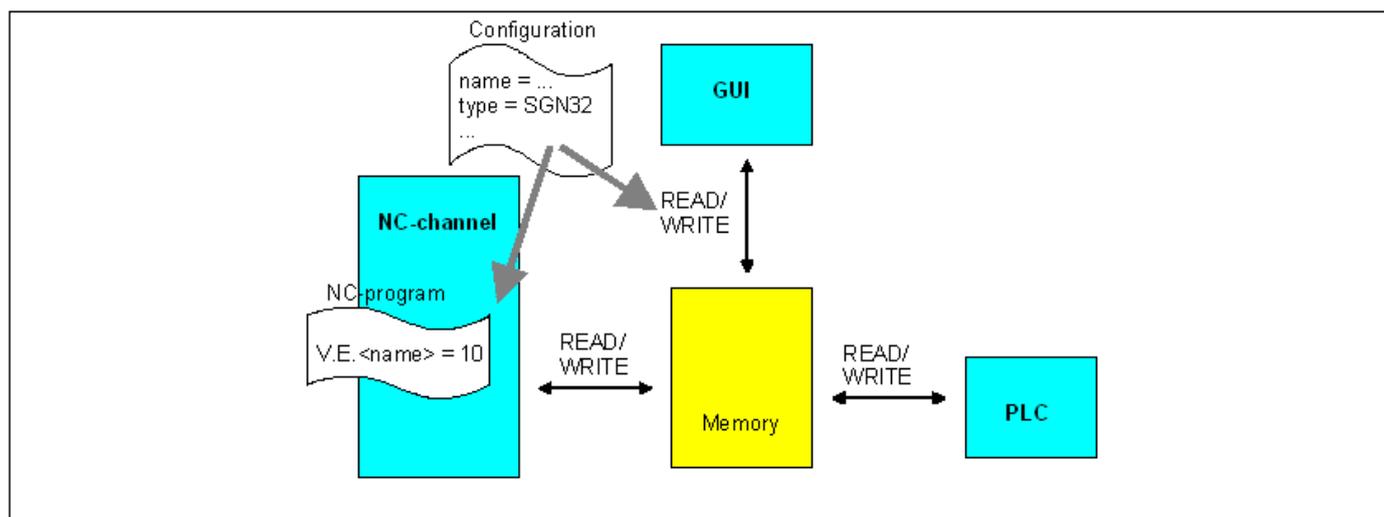


8、V.E.<VAR-NAME>外部变量

概述:

外部变量 V.E...可以在 NC 代码中通过外部地址进行写入，或者通过外部地址进行读取。通过这个变量可以影响 NC 代码的预读，通过 HLI 接口可以进行数据交互。通过 NC 通道访问外部变量能够实现同步、异步访问。外部变量的初始化在系统启动的时候执行。

V.E...外部变量结构



V.E...变量程序示例:

```
N100 $IF V.E.EXT1 >= 100 (Corresponding to the value of V.E.EXT1)
```

(branching off is done into various cases)

```
N110 G01 X100 Y100 F1000
```

```
N120 $ELSE
```

```
N130 G01 X100 YV.E.EXT1 F1000 (Linear interpolation in Y-direction)
```

(with the value from V.E.EXT1)

```
N140 ENDIF
```

```
N150 V.E.EXT1 = V.A.ABS.X (To the external variables the absolute)
```

(X-coordinates are assigned)

```
N160 G01 XV.E.EXT2 (Linear interpolation in X-direction)(with the value from V.E.EXT2)
```

十、NC 代码循环语句

概述:

NC 代码中, 为了方便代码的操作, 提供了丰富的循环指令。这也是倍福 CNC 区别于传统控制系统的特色功能。基于软 CNC 系统, 可以使用类似于 C 语言中的循环指令。包括 IF \WHILE\SWITCH\FOR 等指令结构, 本节介绍这部分指令的使用方法。

常用语法:

1、\$IF 指令

These use of ELSEIF permits :

```

N10 ...
N20 $IF P1 == 0      Only if P1 is equal to 0, the statements N30 to N50
                    will be executed, otherwise the $ELSEIF-condition will
                    be checked, whether P2 >= 0.5 and accordingly N70 to
                    N90 or N110 to N130 will be executed.

N30 ...
N40
N50
N60 $ELSEIF P2>=0.5 The $ELSEIF-condition serves for building up
                    branching nesting into one another.

N70 ...
N80
N90
N100 $ELSE
N110 ...
N120
N130
N140 $ENDIF

```

2、\$SWITCH



Programming example

```

N100 $SWITCH P1=INT [P1*P2/P3] If the result of the arithmetic expres-
sion
                                is equal 1, the blocks after $CASE 1 will
                                be executed (N120 bis N140)

N110  $CASE 1
N120  ... N130
N140  $BREAK
N150  $CASE P2                                If the result is equal to P2, tzhe blocks
                                                N160 ..N170 will be executed.

N160  ...
N170  $BREAK
N300  $CASE n
N320  ...
N330  $BREAK
N350  $DEFAULT                                The $DEFAULT block is optional and serves to
                                                process the NC-blocks.
                                                N360 to N380, if the result of the $SWITCH-
                                                Block has not matched any of the

N360  ...
N370
N380
N390 $ENDSWITCH

```

3、\$GOTO



Programming example

```

%goto

N05 P1=1
N06 P2=1

N10 G74 X1 Y2 Z3
N11 X0 Y0 Z0

N15 $IF P1==1 $GOTO N40: -> Jump from outside to N40 into a control
      block

N20 X10
N25 Y10

N30 $IF P1==2
N35 X20
N40: $IF P2==1
N45 X30
N50: Y30 $GOTO N65: -> Jump to N65 between control block levels
      (IF-ELSE)

N55 $ELSE
N60 Y40
N65: X40
N70 $ENDIF

N80 Z99

N999 M30

```

4、\$FOR



Programming example

```

Negative step width:
N100 $FOR P1= 100, 10, -2 P1 is pre-allocated with 100 at the loop
      beginning. The counting loop is traversed till
      P1 has fallen below the value 10, whereas P1 is
      decremented by 2 at the end of every loop pass.
      Within the counting loops the NC blocks N110 to
      N130 are executed.

N110 X SIN [P1 * 5]

N120 Y COS [P1 * 5]
N130 ...
N150 $ENDFOR

N100 $FOR P1= 100, 10, 1 P1 is pre-allocated with 100 at the loop
      beginning. The counting loop is traversed till
      P1 has exceeded the value 10, i.e. only once.
      Within the counting loops the NC blocks N110 to
      N130 are executed.

N110 X SIN [P1 * 5]

N120 Y COS [P1 * 5]
N130 ...
N150 $ENDFOR

Endless loop
N100 P2=20
N110 $FOR P1= 100, 10, 0 Endless loop
N120 $IF P2 == 50
N130 $BREAK
N140 $ENDIF
N150 $ENDFOR

```

5、\$WHILE



Programmierbeispiel

```

N90 P1 = 100.0
N100 $WHILE P1 > 0.5      P1 > 0.5 is tested for FALSE at the loop
N110 P1 = P1 - 1.5 Y P1  beginning. The loop is traversed till P1 ful-
N120 $ENDWHILE          fills the aborting condition
N130 ...

```

6、\$DO

7、\$REPEAT



Programming example

```

N10 X0 Y0 Z0
N20 P2=10 P1=0
N30 $DO
N40   P1=P1+1
N50   XP1
N60 $ENDDO P1 <= P2      P1 is tested for FALSE at the end of the loop.
                          The loop is passed till P1 does not fulfill the
                          condition any longer.

N99 M30

N10 X0 Y0 Z0
N20 P2=10 P1=0
N30 $REPEAT
N40   P1=P1+1
N50   XP1
N60 $UNTIL P1 > P2      P1 is tested for TRUE at the end of the loop.
                          The loop is passed till P1 does fulfill the
                          condition.

N99 M30

```

8、\$BREAK



Programming example

```

N10 $WHILE <expr1>      The loop is terminated if
N20 ...                expr1 "not valid" or
N30 ...                expr2 "valid"
N40 $IF <expr2>
N50   $BREAK
N60 $ENDIF
N70 ...
N80 ...
N90 $ENDWHILE
N100 ...

```

9、\$CONTINUE



Programming example

```
N10 $FOR <expr1>
N20 ...
N30
N40   $IF <expr2>
N50     $CONTINUE
N60   $ENDIF
N70 ...
N80
N90 $ENDFOR
N100 ...
```

N70 and N80 are executed if
expr2 is "not valid"

附录一：

CNC 软件更新

CNC 部分版本在应用中需要更新补丁，譬如我们在使用 CNC 1506E 版本的时候，如果用到转为激光切割开发的尖叫控制功能时，需要加装补丁文件。

更新补丁方法如下：

1、解压缩 TcISG 1506e.01 文件。



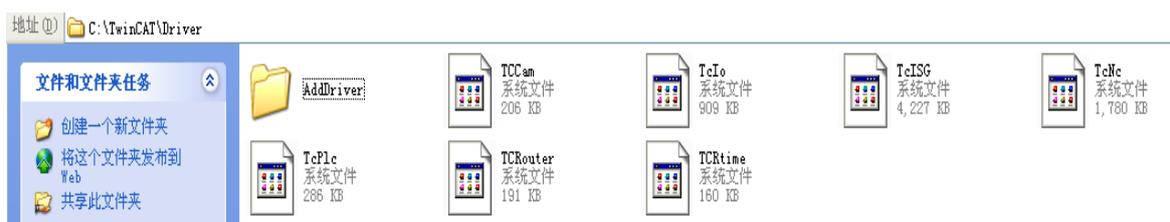
2、将解压文件夹中的【ahmi_ads.exe】复制到 TWINCAT 软件安装位置的 C:\TwinCAT\Cnc Tools。

将压缩文件内容替换到 C:\TwinCAT\Cnc Tools 中的【ahmi_ads.exe】文件。



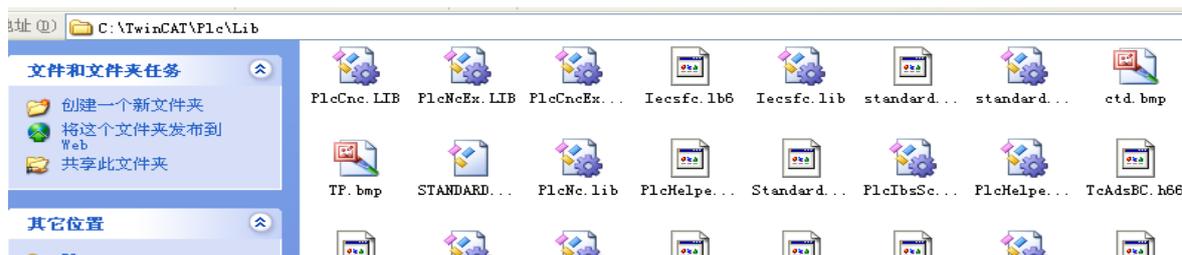
3、将解压文件中的【TcISG.sys】复制到 TWINCAT 软件位置 C:\TwinCAT\Driver 位置。

将压缩文件内容替换到 C:\TwinCAT\Driver 中的【TcISG.sys】文件。



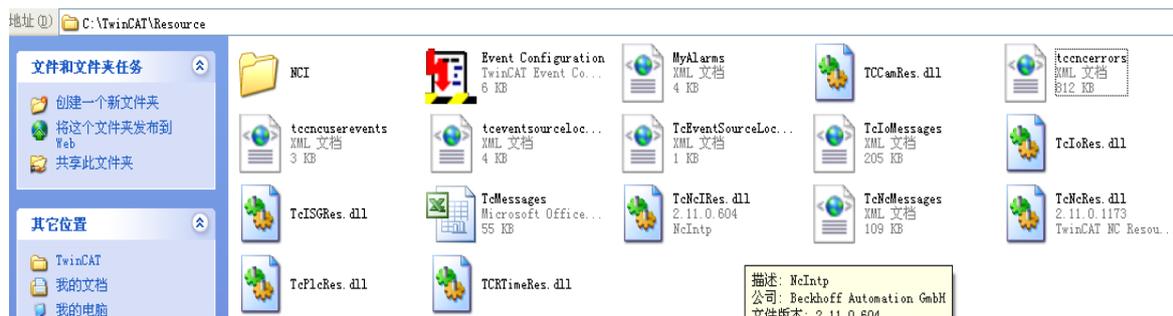
4、将解压文件夹中的【***.lib】文件复制到 TWINCAT 软件安装位置的 C:\TwinCAT\Plc\Lib。

将解压文件夹中的【***.lib】文件全部复制到 TWINCAT 软件安装位置的 C:\TwinCAT\Plc\Lib。



5、将解压文件中的【tccncerrors】复制到 TWINCAT 软件安装位置的 C:\TwinCAT\Resource。

将解压文件夹中的【tccncerrors】文件全部复制到 TWINCAT 软件安装位置的 C:\TwinCAT\Resource。



6、将以上内容复制替代完成后，重新启动 TWINCAT 软件。

7、完成 CNC 文件更新！

附录二：

HMIPRO 界面介绍

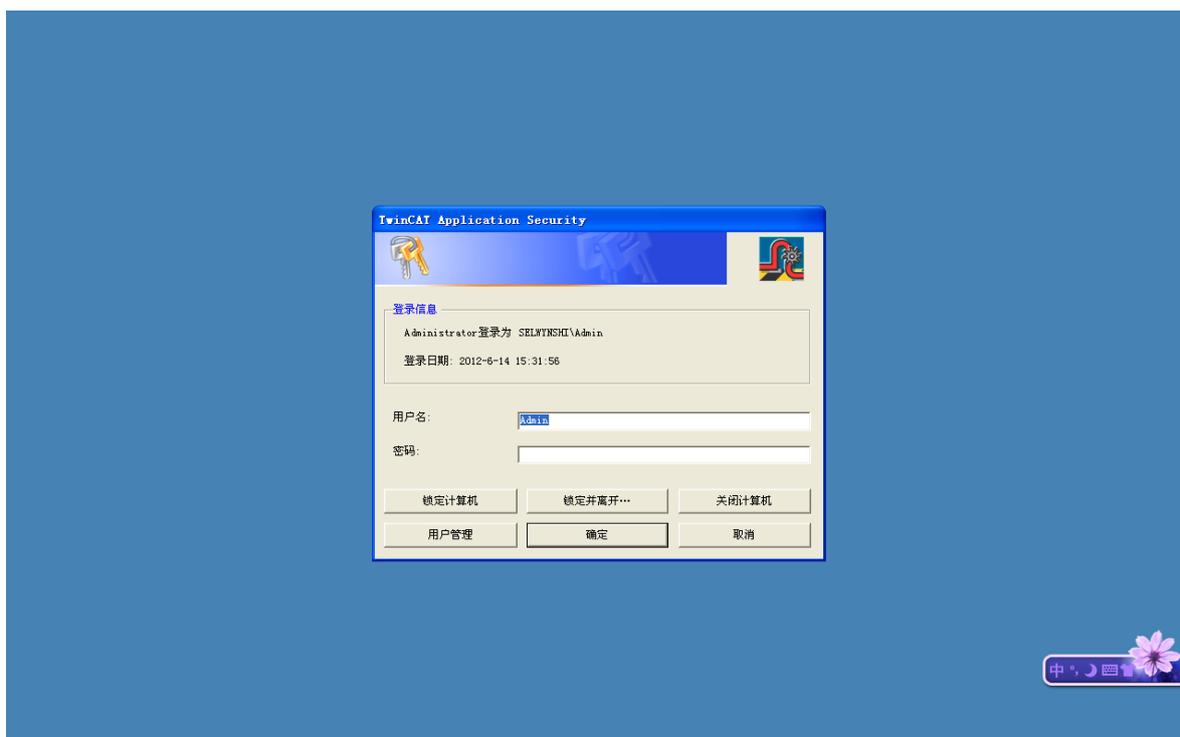
HMIPRO 界面是倍福基于 C#语言基础上开发的成熟的界面，具有画面美观、操作简单实用、功能全面等优点。倍福对于该软件是全面开放，代码公开，便于客户自主修改。

系统画面

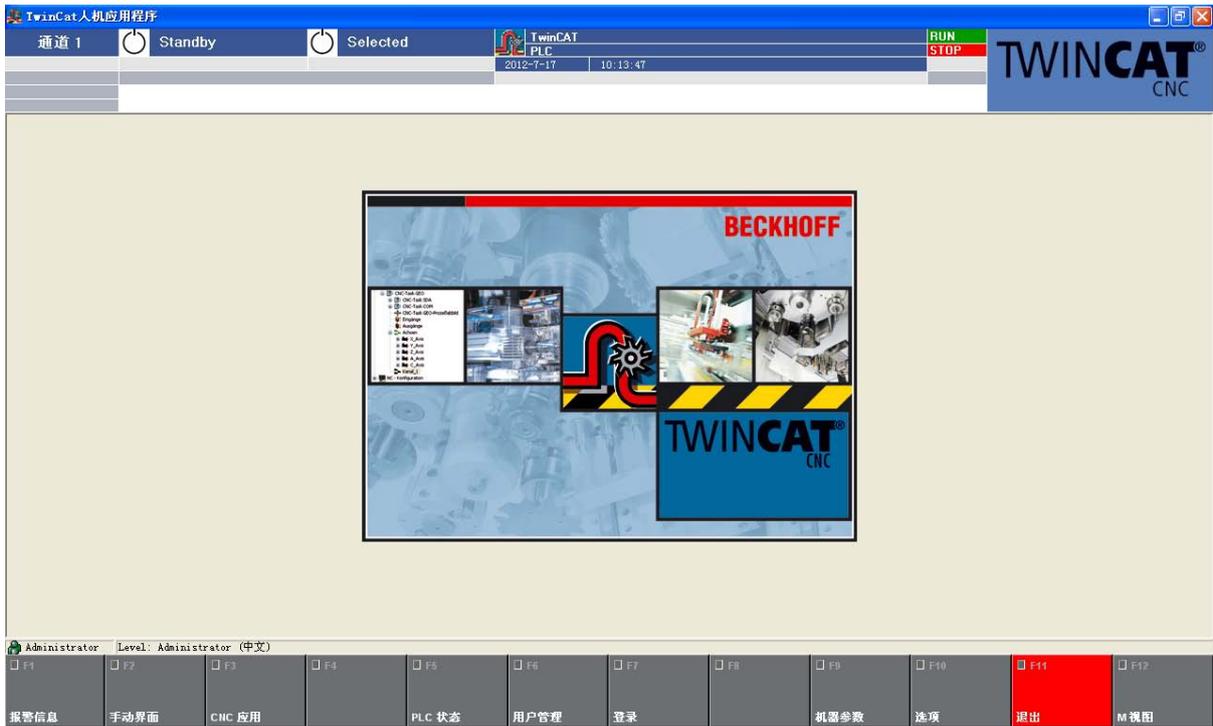
1、登陆界面

- 工控机电脑电源启动后，自动登陆置 WINDOWS 开机画面，若需密码登陆，则输入密码：1；
- 登陆 windows 之后，进入如下画面，登录用户名为：Admin；密码为空；

画面如下：



- 点击确定进入画面，该画面为 CNC 系统的主画面，下面的按键（F1-F12）对应于键盘的功能键 F1-F12；



报警信息 F1 键：可显示倍福的 CNC 系统报警信息以及报警记录，具有历史报警信息记录等功能；

CNC 应用 F3 键：倍福 CNC 系统操作主画面，包含 CNC 操作信息以及参数、跟踪界面以及手动功能等；

PLC 状态 F5 键：监控 PLC 变量，当前显示数值，包括 bool、int、lreal 等数值类型；

机械参数 F6 键：用于设置机械参数，如手动速度、机械位置参数等；

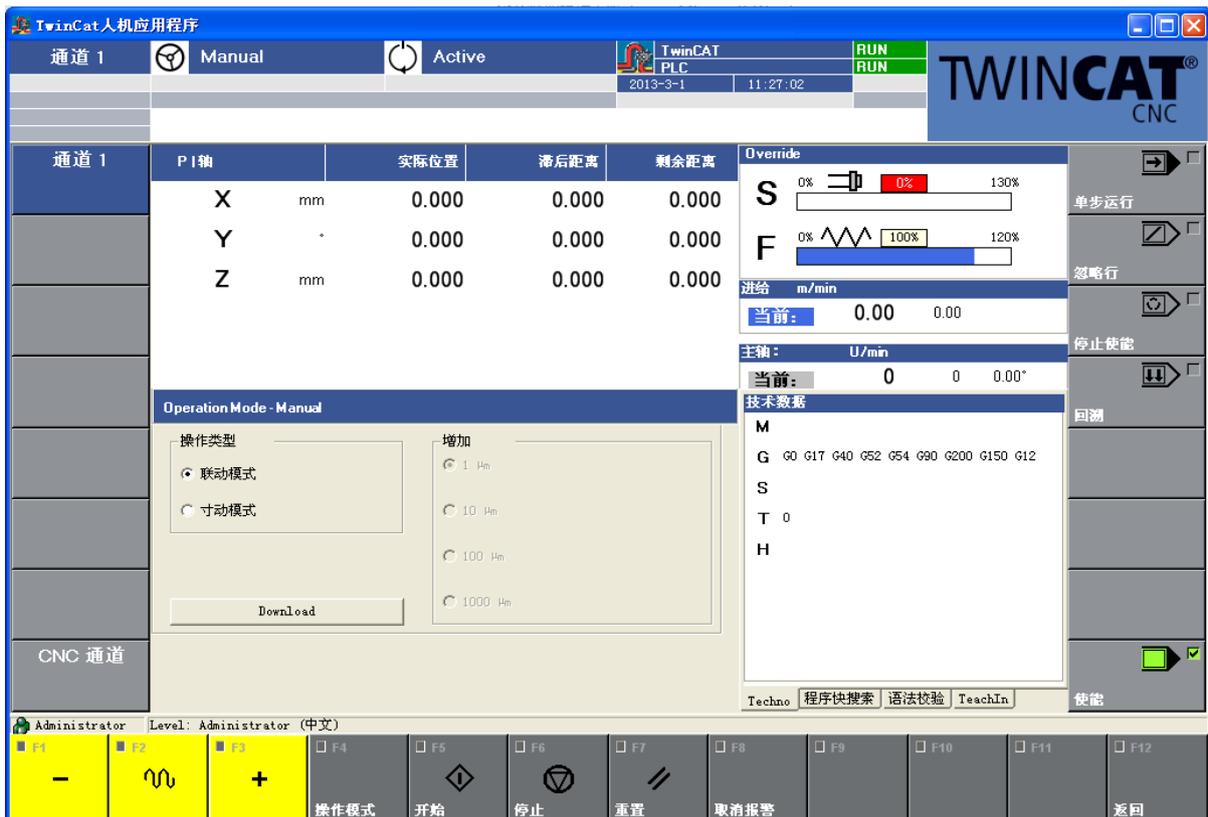
登陆系统 F8 键：用于切换用户权限；

用户管理 F9 键：此处可以划分系统使用者即用户等级，可以对机械参数中的数值修改权限设定，用于某些固定参数或者系统参数的保护；

选项操作 F10 键：用于语言切换即 CNC 内部系统参数修改；

退出操作 F11 键：退出 HMIPRO 界面；

- ✓ CNC 应用 F3 键，显示如下图示：



2、功能显示:

界面中实时显示当前 X\Y\Z 轴当前位置、滞后距离以及剩余运行距离，并能够实现机械坐标系以及工件坐标系的切换、并可以当当前的之后位置改为终点位置、激活位置等。

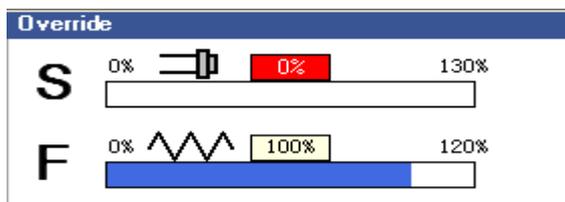
(了解：在之后距离处单击右键并选择坐标系即可，也可切换显示信息)；

PI轴		实际位置	运行距离	剩余距离
X	mm	0.000	<input checked="" type="checkbox"/> 工件坐标系 <input type="checkbox"/> 机床坐标系	0.000
Y	mm	0.000	<input checked="" type="checkbox"/> 滞后距离 <input type="checkbox"/> 终点位置 <input type="checkbox"/> 剩余距离 <input type="checkbox"/> 激活位置	0.000
Z	mm	0.000		

3、功能操作：显示当前的机床的工作倍率，并可通过按键进行倍率加减操作，

S: 主轴倍率数值：最大范围 0%--130%；

F: 进给轴倍率数值：最大范围 0%--120%



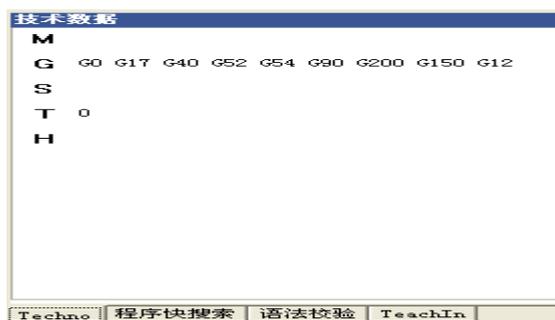
4、进给速度显示：显示 X、Y 轴当前合成路径速度（单位 M/MIN）；

进给	m/min	
当前:	0.00	0.00

5、主轴速度、位置显示:

主轴:	U/min		
当前:	0	0	0.00°

6、功能显示：显示当前运行 M、G 代码信息；



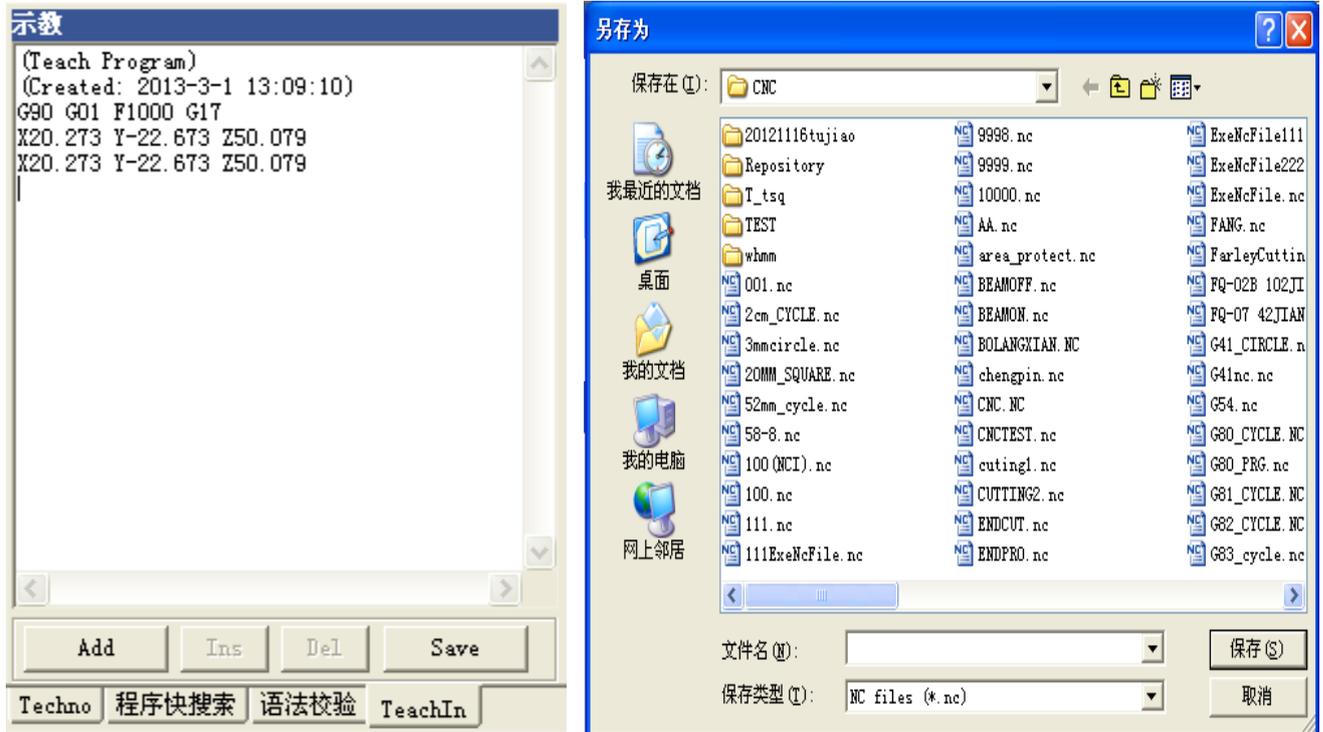
7、示教功能：

示教位置：

手动移动 XYZ 轴到目标位置，按 Add 按键将当前位置保存在示教表格中，从下图中可以看到，与位置一起保存的内容包括 G90 G01 F1000 G17，这是默认的运动模式。

另存位置：

将示教位置保存完毕之后，单击 save，实现 G 代码保存，以 .NC 文件名保存。保存完成之后，可以对该文件进行处理。



8、功能操作：系统伺服驱动设备使能上电；



9、单步执行：选中单步按键后，按下开始之后，G 代码执行一行。

回溯执行：选中回溯按键后，按下开始按键，G 代码将从当前位置回退，直到按下 STOP 按键或者回退到程序起始行。



10、操作按键功能说明：



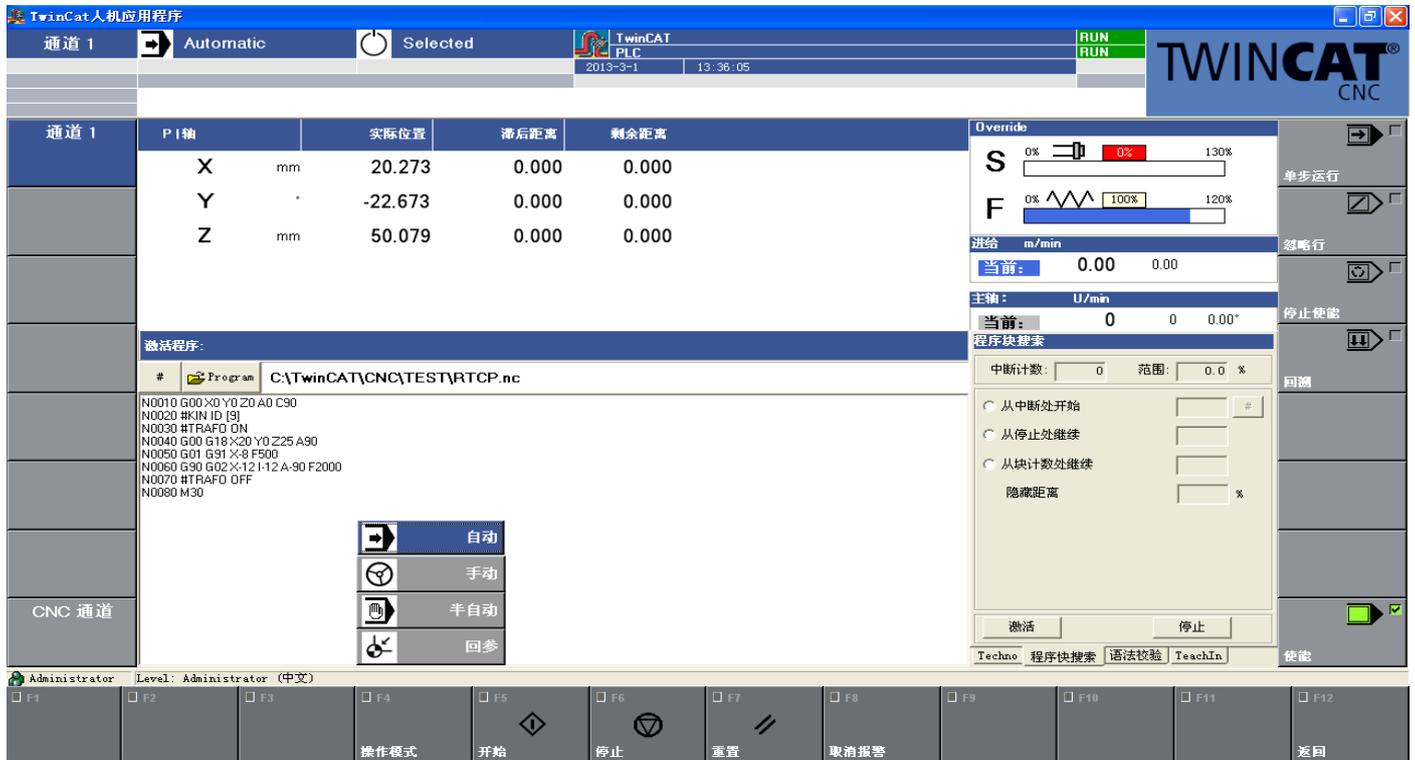
- 操作模式选择 F4 键：
用于选择操作模式包括自动、半自动、手动、回参；如右图所示：
- 开始按键 F5：
用于自动模式以及半自动模式时 G 代码运行执行操作；
- 暂停按键 F6：
用于自动模式以及半自动模式时 G 代码暂停运行操作；
- 复位按键 F7：
用于所有未实现的系统复位，此时对于系统错误以及 PLC 错误及程序进行复位操作
注意：在执行 G 代码之前需要按复位键以确保清除上次执行 G 代码；
- 报警确认 F8：
用于在自动运行模式下，对于 PLC 中的故障信息的报警再确认操作；
- 返回按键 F12：
返回。



系统操作:

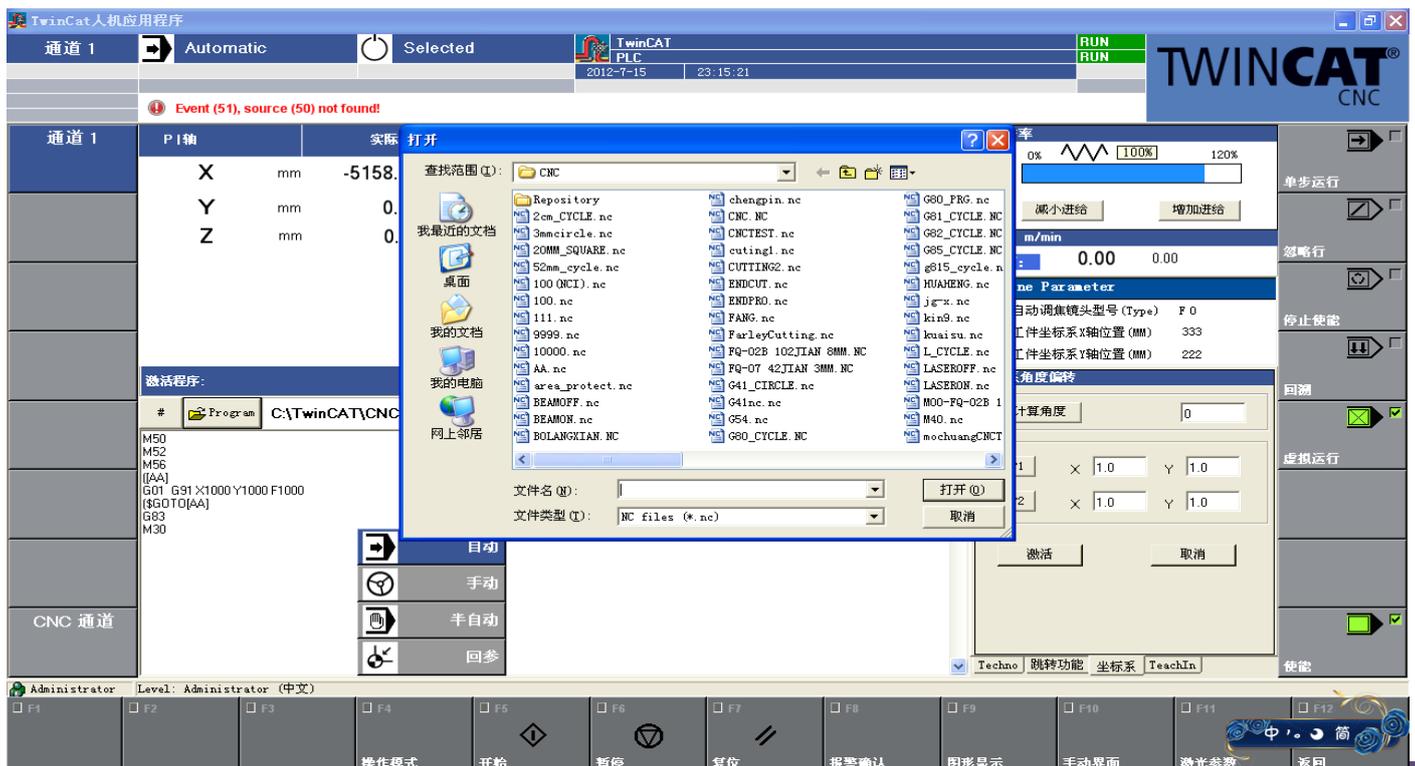
自动模式:

G 代码的运行必须在这个模式下, 如下图, 运行时需选择开始按键



1) 选择 G 代码文件:

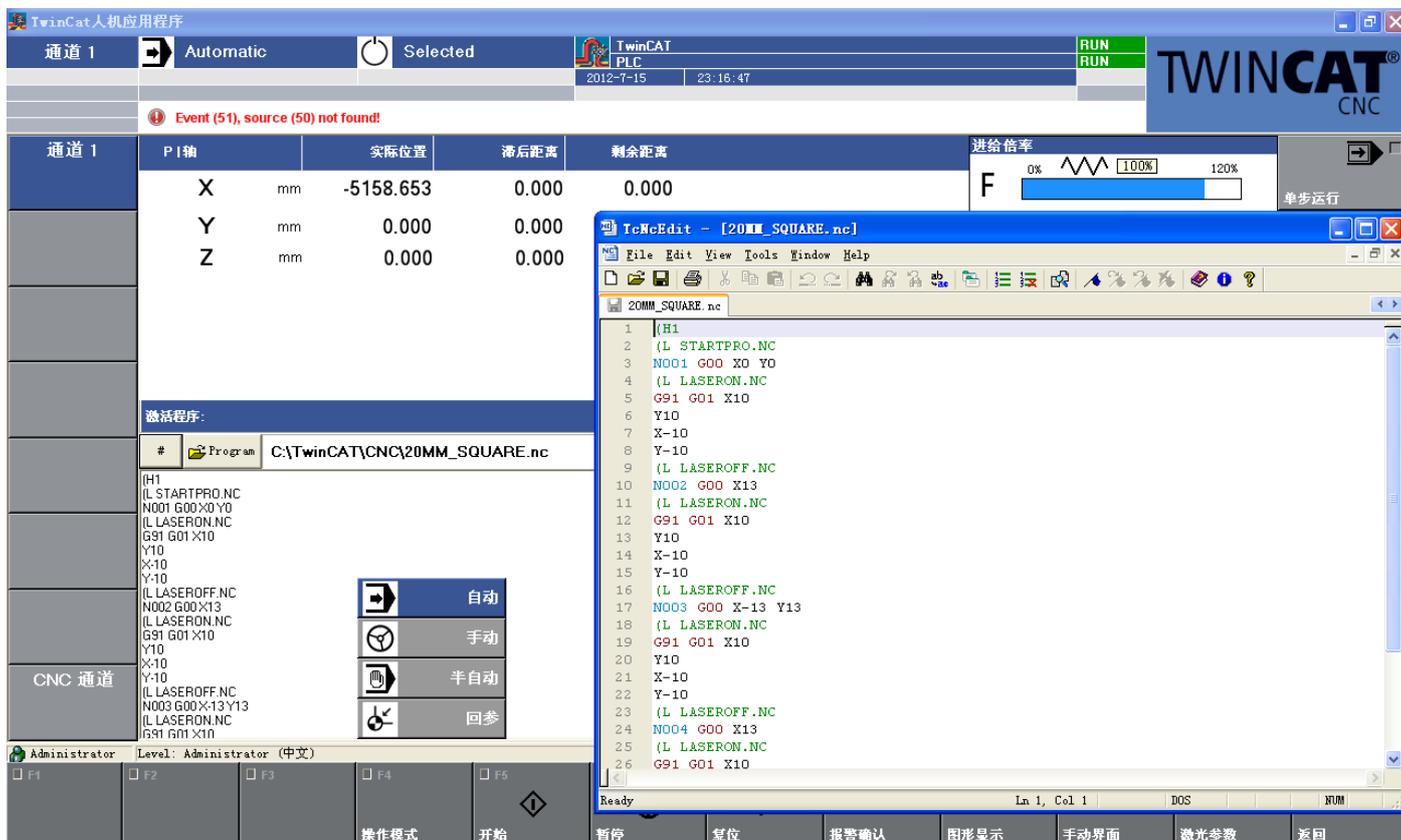
G 代码文件保存路径: C:\TwinCAT\CNC (将 G 代码拷贝到该路径下);



2) 编辑当前 G 代码

单击 ‘#’ 键，编辑当前 G 代码文件，保存。

注意：此时需要重置才能实际应用到 G 代码文件。

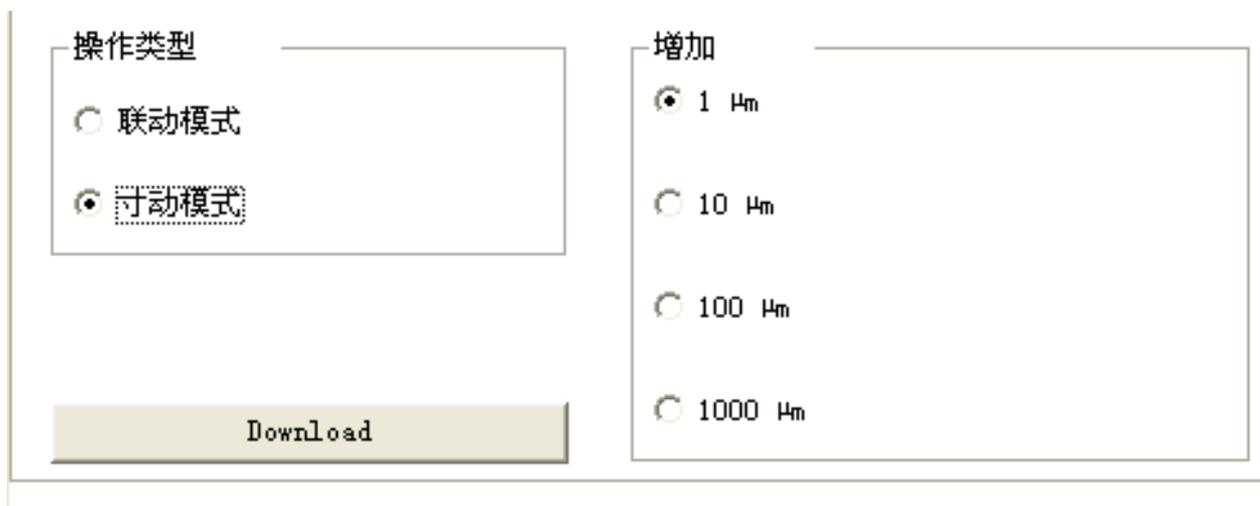


手动模式：

1) 模式选择：联动模式以及寸动模式切换。

联动模式：操作任意一个轴，按键保持即可实现该轴的连续运行；

寸动模式：操作任意一个轴，可以按照右边列出的位置进行点动；



2) 快慢速选择:

正负向软按键移动，中间可以快慢速切换。此时可通过外部按键进行手动操作可使用快慢速切换。



快慢速设置画面在机械参数中设置 X、Y、Z 三个轴的位置。

3) 手动操作界面

通道 1 Manual Active TwinCAT PLC 2012-7-15 23:20:41 RUN RUN TWINCAT CNC

Event (51), source (50) not found!

通道 1	P I 轴	实际位置	滞后距离	剩余距离
	X	mm -5158.653	0.000	0.000
	Y	mm 0.000	0.000	0.000
	Z	mm 0.000	0.000	0.000

进给倍率 F 0% 100% 120%

进给 m/min 当前: 0.00 0.00

Machine Parameter

自动调焦镜头型号 (Type) F 0
工件坐标系 X 轴位置 (MM) 333
工件坐标系 Y 轴位置 (MM) 222

技术数据

M
G 00 G17 G40 G52 G54 G90 G200 G150 G12
S
T 0
H

Administrator Level: Administrator (中文)

操作类型: 联动模式 寸动模式

增加: 1 Hz, 10 Hz, 100 Hz, 1000 Hz

Download

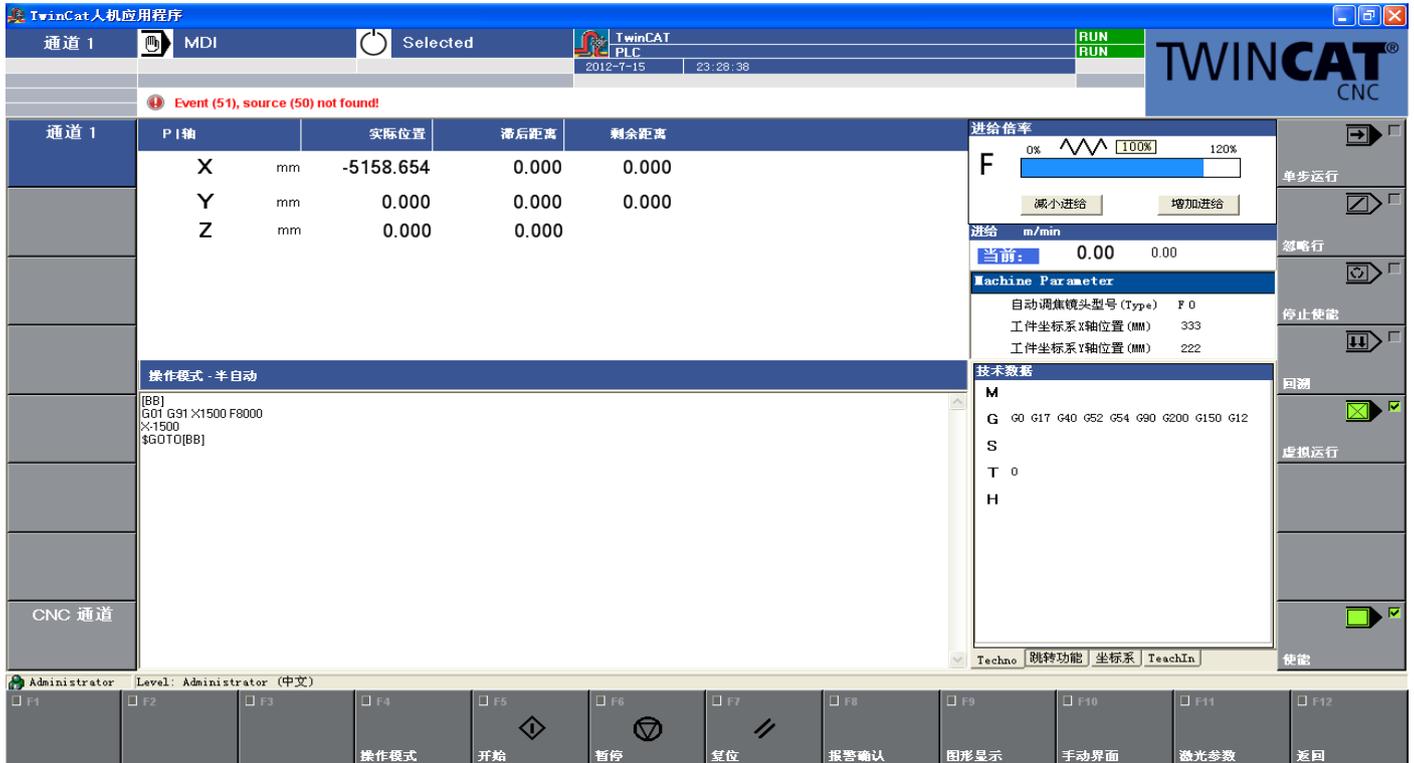
Techno 跳转功能 坐标系 TeachIn 使能

F1 - F2 ~ F3 + F4 F5 F6 F7 F8 F9 F10 F11 F12

操作模式 开始 暂停 复位 报警确认 图形显示 手动界面 激光参数 返回

半自动模式:

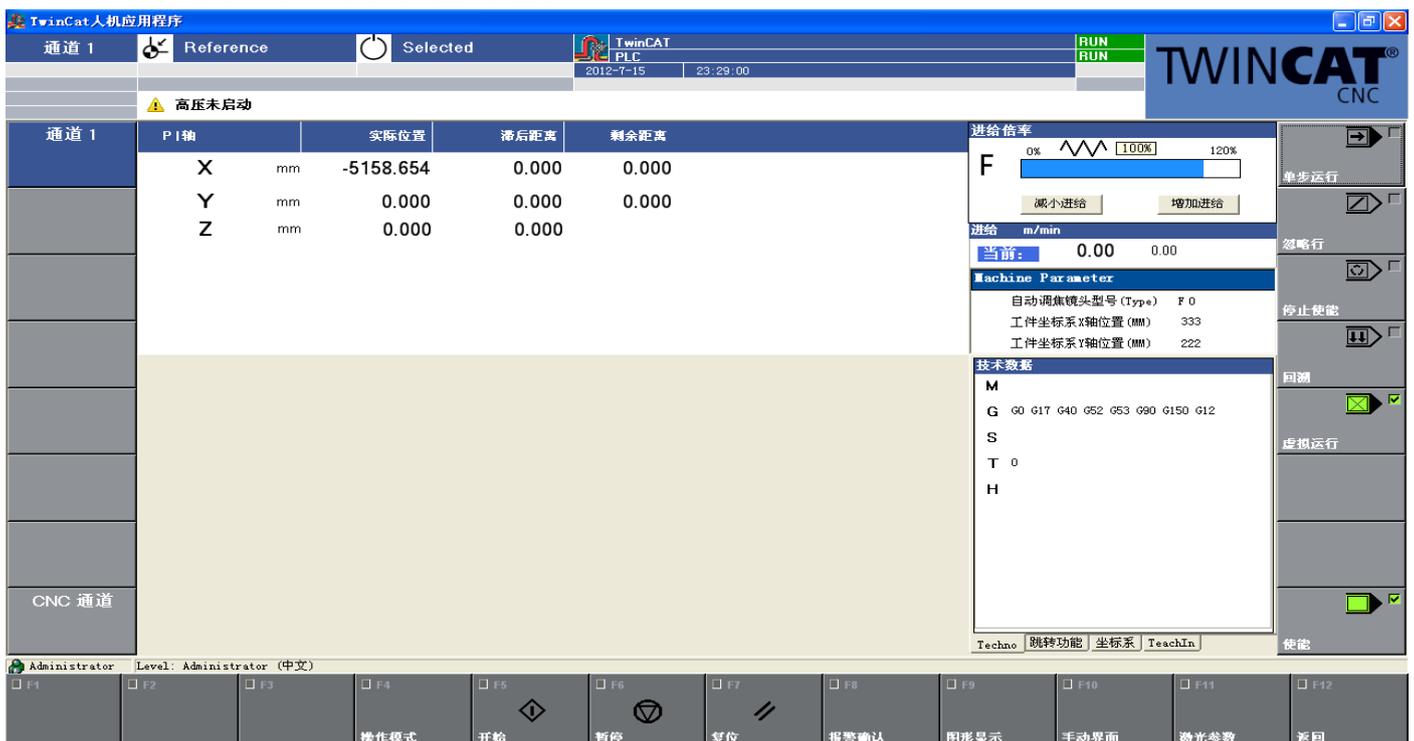
此界面可以直接输入 G 代码以及 M 函数来实现直接运行，按开始按钮运行 G 代码或者使用暂停按钮停止运行 G 代码；



回参模式:

按回参按钮，以及开始按钮可以执行 X、Y、Z 轴的回参操作，先完成 Z 轴以及调焦镜头回参再执行 XY 轴；
注意：回参完成之后，X、Y 轴显示出现绿色，表明已经回参完成；

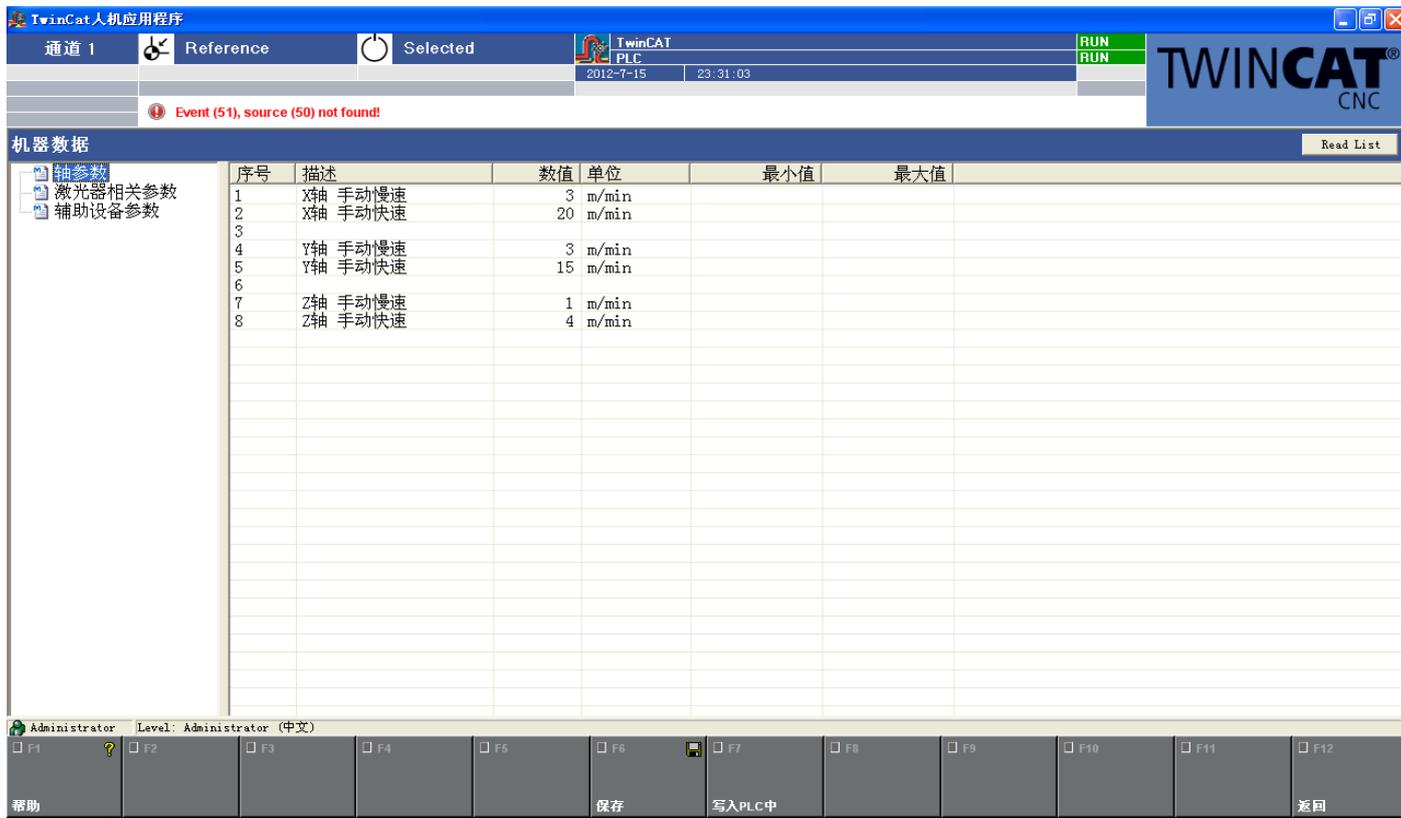
回参完成之后，软限位才能够有效！



机械参数

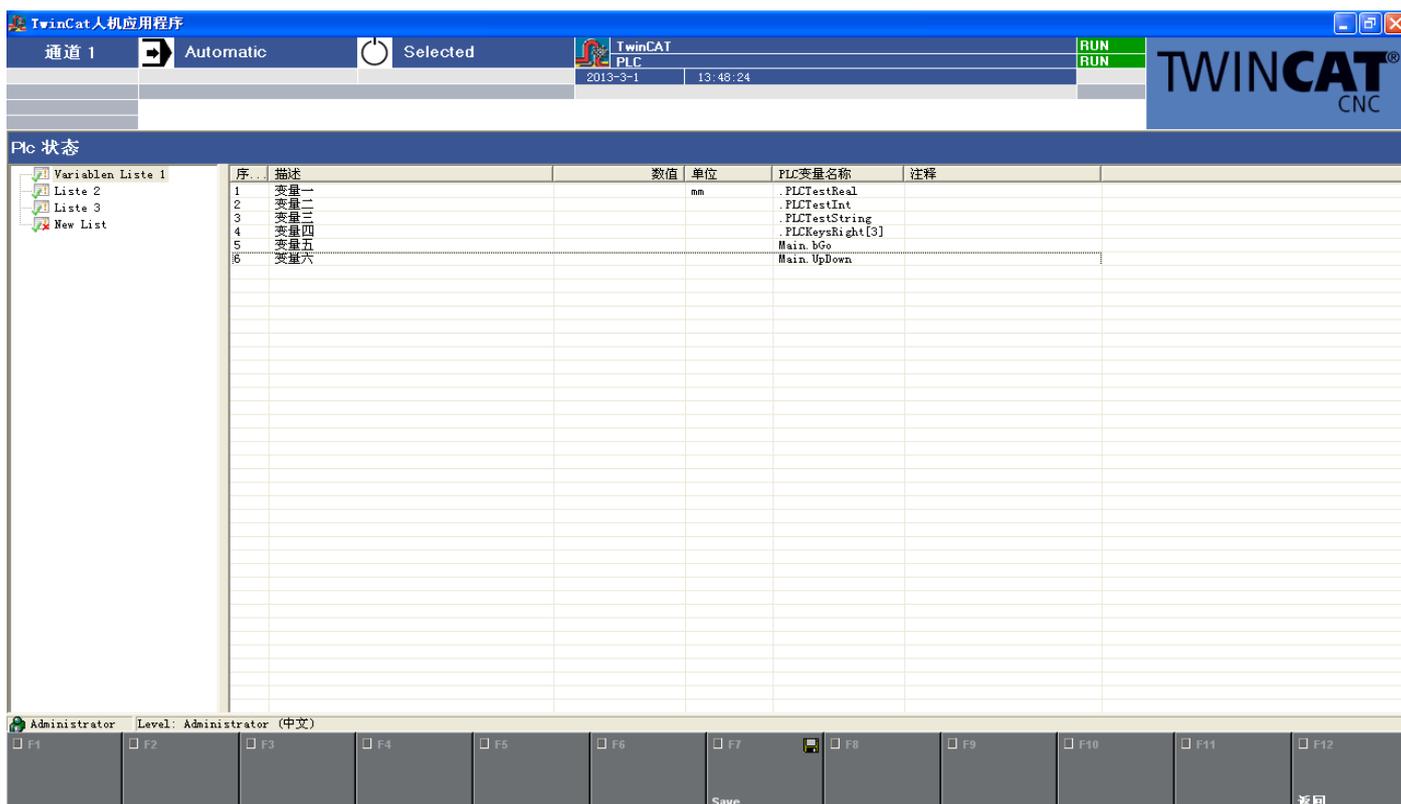
在机械参数中，可以设置轴参数设置、工艺参数等；

单击主界面的机械参数按钮，进入该画面，通过数组的方式与 PLC 中变量进行连接，从而可以修改工艺参数。



监控参数:

单击主界面的 PLC 状态，进入该画面，修改变量名称与 PLC 变量名相同，实现变量实时监控。



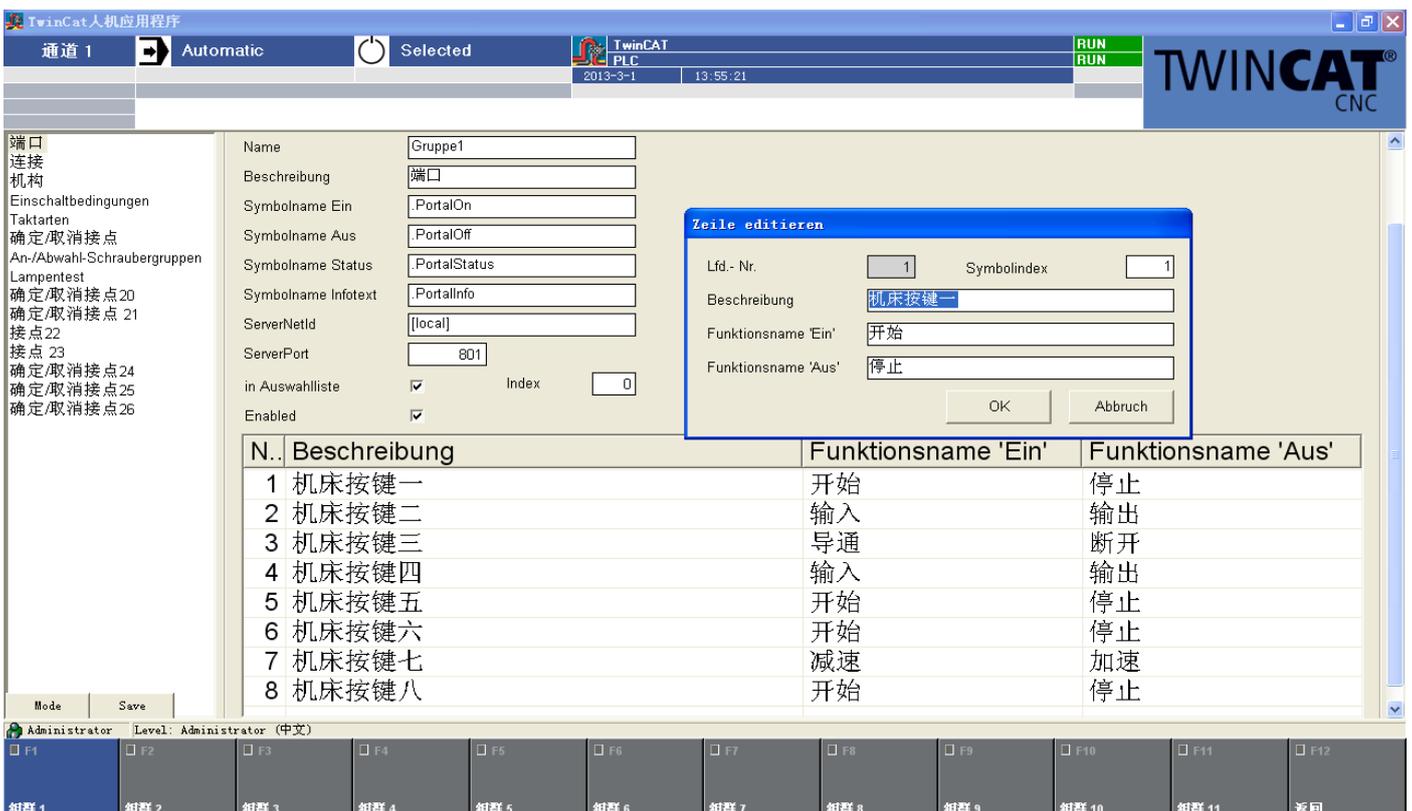
手动画面

在 HMIPRO 中，可以设置一些软开关。单击手动操作，进入下列界面，通过数组的方式，对 PLC 中的变量进行操作。同时，可以对变量的内容进行修改，并有夜色变化，表示状态改变。



单击画面中的 mode 按钮，进入下图，在此图中修改按钮信息。

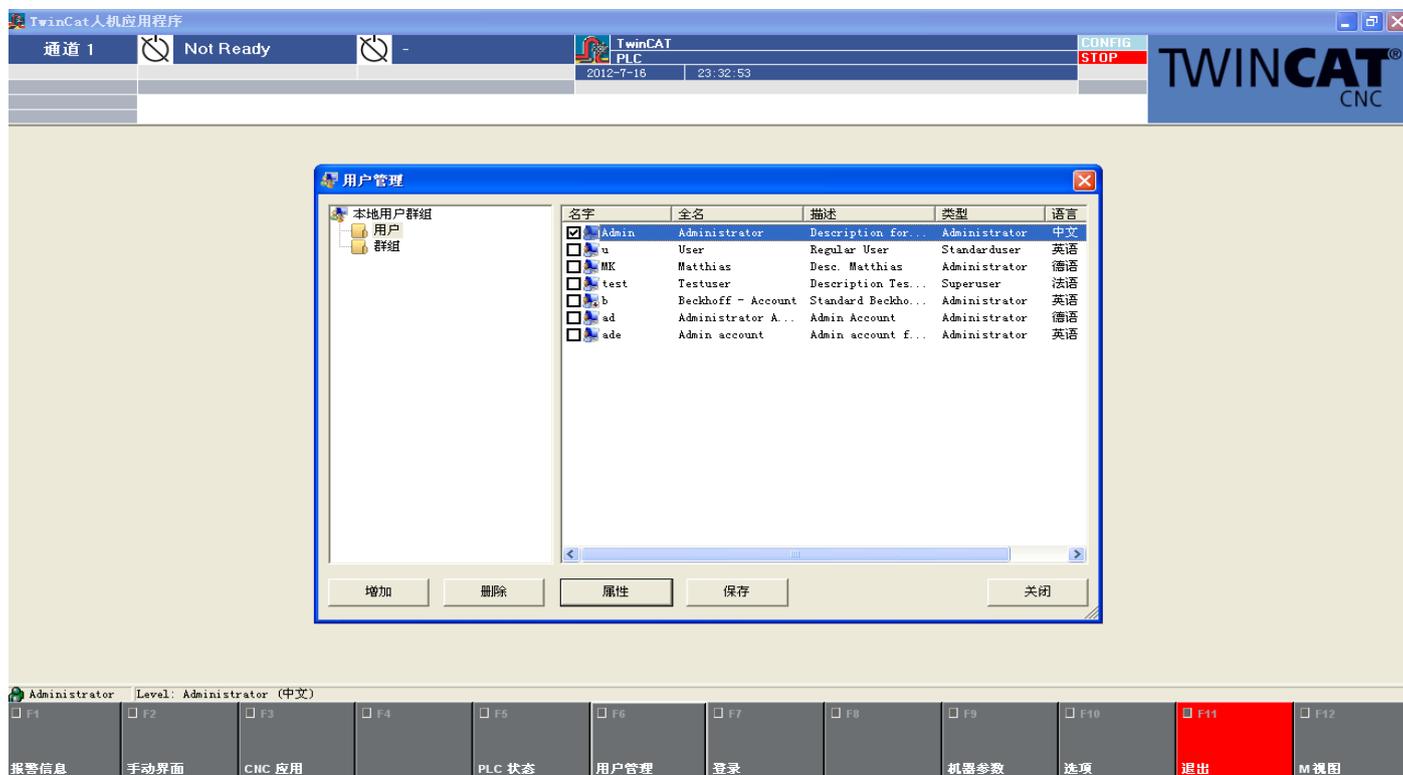
在下图中可以看到 PortalOn[i]代表左边打开按钮，PortalOff[i]代表右边关闭按钮；PortalStatus[i]表示左右两侧按钮的状态，0,1,2,3 代表状态值，用于改变按钮颜色。



用户管理

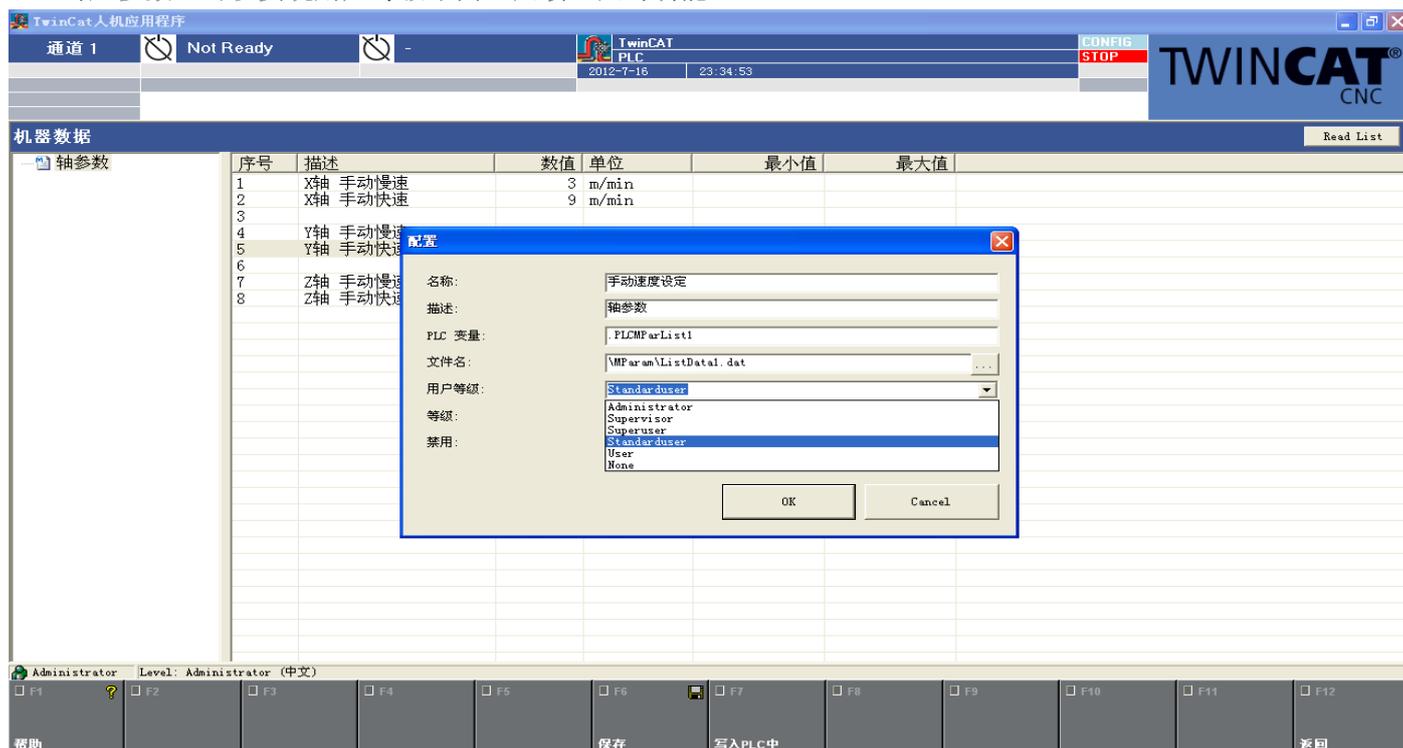
用户账号设定

添加用户名，在属性中修改账户名称、密码、启动语言种类、所属级别；



应用参数使用用户级别

对应参数，可以实现用户等级不同，只读，只写功能。



切换账户

单击登陆，进入如下界面，可以通过用户名密码进入不同的账户。



附录三

报警信息编辑操作

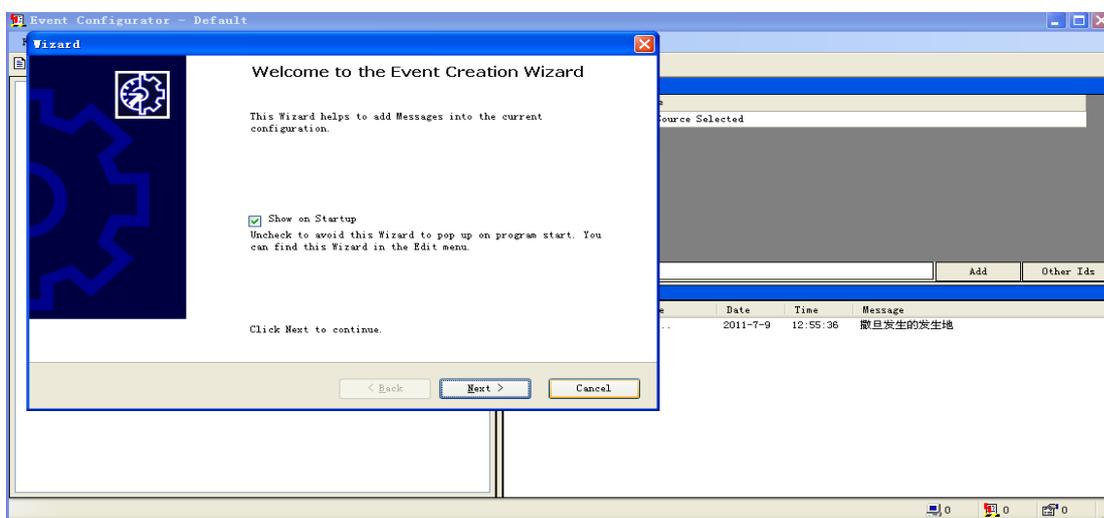
报警信息自定义,通过 TwinCAT 自带的 eventlog 定义,并显示在界面中,此部分内容将结合 PLC 程序进行讲解。此处列举如何在 eventlog 软件中配置报警信息。

1、设置 Event Configuration 文件

(防止文件丢失破坏建议保存在 C:\TwinCAT\Resource\)

Event Configuration 文件位置: C:\TwinCAT\EventLogger\EventConfigurator

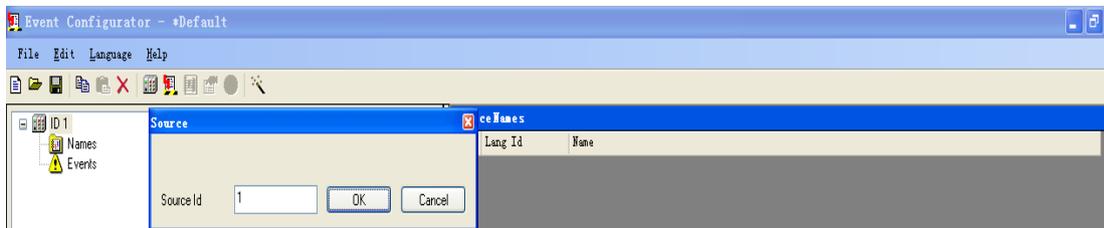
➤ 打开 Event Configuration 文件,配置报警信息内容;



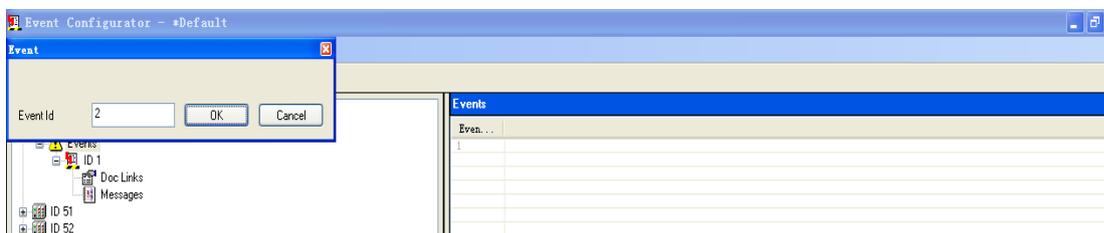
➤ 点击 CANCEL,进入如下画面, 点击  如下图;



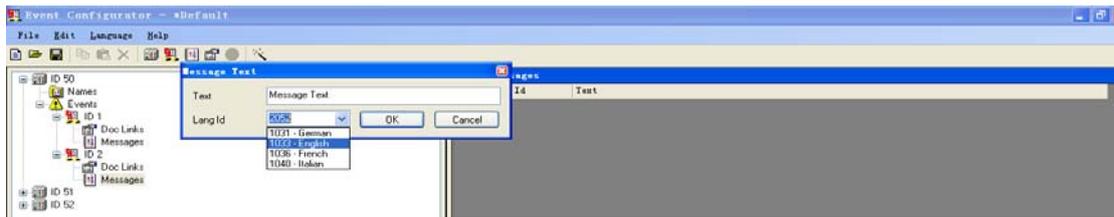
➤ 修改 sourceID,改为 50、51、52 可以继续添加的;



➤ 添加 EVENT ID(1-N);

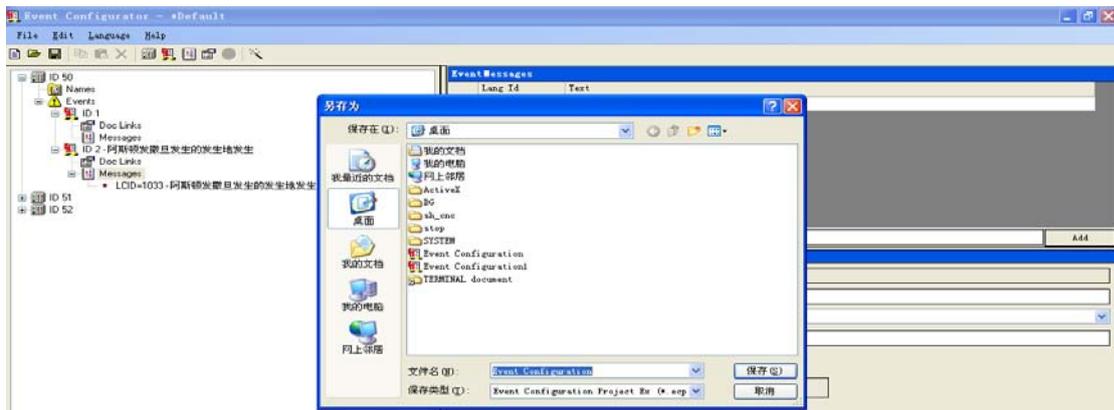


- 添加 MESSAGE,选择语言为 1033, 在 HMIPRO 中默认的语言是 1033 英语; 修改内容的话, 需要先 APPLY,然后再激活!



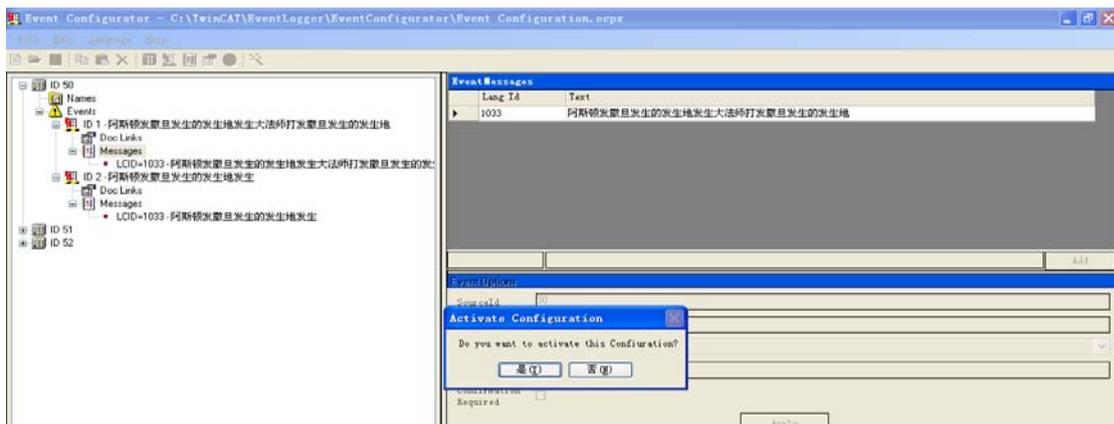
2、保存配置的报警文件

修改完成之后, 即将想要添加的报警信息添加完成之后保存这些报警信息, 被下次自动启动 (防止文件丢失破坏建议保存在 C:\TwinCAT\Resource\);



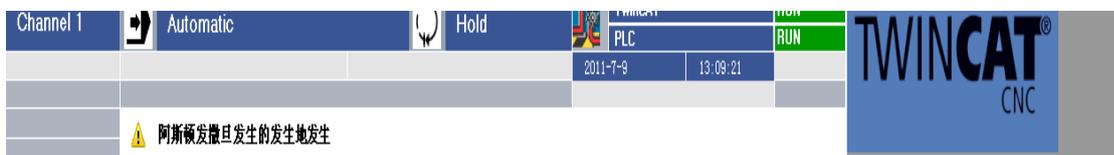
3、激活配置的报警信息

保存完成之后, 弹出对话框是否激活这些配置, 点击确定;



4、显示报警信息

重新启动 PLC 以及配置文件之后, 可以发现错误信息报警



附页四：德语对应翻译

通道参数

achs_anzahl	轴总数
achse	轴
bezeichnung	标示
einrechnen_mit_t	把 t 算入在内
einzelstschrittmodus	单步模式
feed_to_weackest_axis	进给到最弱的轴
fehler	错误
fehler_gesamt	错误_总合
gruppe	组
interaktiv	交互
kasto_reject_rest_block	kasto_拒绝_剩余_块
lin_aufloes	线性_分辨率
makro	宏观
mass_einh	质量_单位
messtyp	测量类型
mittelpkt_diff	中点(平均)_差值
mittelpkt_faktor	中点(平均)_因子
param_haltend	参数_保持
pro_zeile	每行
prozess	进程
residual_path	残余_路径
spind_aufloes	轴_分辨
synchro_data.koppel_gruppe[i].paar[j].	同步_数据.耦合_组[i].对[j]
track_deviation	轨迹偏差
vector	向量
versch_im_durchm	(在测量中的偏移)
vorhanden	存在的
wrk_im_kanal_vorhanden	wrk_在通道中存在
zeit	时间

轴参数

beschl	加速
brems	减速
abs_pos_gueltig	绝对_位置_有效
abs_pos_gueltig	绝对_位置_有效
achs_mode	轴模式
antr.sercos.	sercos 驱动
losekomp	偏差值
anz_driftwerte	调出
anzeige	显示
backlash	反向间隙

bahnregelung	轨道控制
bahnrg	轨道控制
beschl_kennlinie	加速曲线
beschl_zu	加速_到
beweg_richt	运动方向
br_p_nenner	轨道控制_p_分母
daempfung	阻尼
drift_wert	漂移值
dynamik	动态
eigenfrequenz	固有频率
eilgang	快速运动
factor_denominator	因子_命名
feed_const_denom	进给_常数_命名
feinaufloesung	高分辨率
feininterpolation	精细插补
filter_zeit	滤波器_时间常数
geom	几何
geschw	速度
getriebe	驱动装置
grenz	边缘的
grenz_stufe	极限_等级
gueltig	有效的
handbetrieb	手动驱动
kenng	核心参数
konst	常量
laenge	长度
mess_neg_flanke	测量下降沿
mess_signal_achs_steu	测量信号轴控制
mess_signal_taster	测量信号按钮
messachse	测量轴
messtaster	测量按钮
mod_rechn	mod_计算
modulo_fehler	模_错误
modulo_umdreh	模_转动
nenner	分母
nocken_laenge	凸轮_距离
nummer	编号, 数字
offsetgrenze_neg	偏移极限_负
pos_einschw_zeit	位置_(到达)_时间
pos_refpkt	位置参考点
prozent	百分比
ref_ohne_nocken	回参_无凸轮
ref_richt	回参_方向
regelgrenze	控制极限
schalt_pos	切换位置
slep_ueberw_typ	slep_监视_类型
stufe	等级; (发展的) 阶段
ueberwach	监控
vorsteuer	预控制器

vorsteuerung	预控制, 预调
richtung	方向
wegaufn	路径接受
wsi_meldung	wsi_通知
zeitkonstante	时间常数

附页五：数控机床常用术语

数控机床技术常用术语

为了方便读者阅读倍福 CNC 资料与快速入门文档，在此我们选择了常用的数控词汇及其英语对应单词，帮助读者更好地理解相关内容。

1) 轴 (Axis)

轴是指机器的一个独立运动。通常，一台数控机床由多个轴组成。

2) 主轴 (Spindle)

直接用于加工，且加工时必须运动的轴。一台机床可以有多个主轴，其中最常用的一个称为主要主轴 (Main Spindle 或 Master Spindle)。主轴通常为旋转轴。

3) 进给轴 (Feed Axis)

直接驱动刀具或工件移动的轴。一台机床通常有多个进给轴，例如，车床有 X、Z 轴，铣床有 X、Y、Z 轴。

4) 几何轴 (Geometry Axis)

直接组成右手几何坐标系的进给轴。例如车床几何轴有 X、Z 轴，铣床、镗床几何轴有 X、Y、Z 轴。

5) 辅助轴 (Accessorial Axis)

机床中用于辅助运动的轴，如刀库、机械手、附件库控制轴等。

6) 直线轴 (Linear Axis)、

运动轨迹是直线的轴。多数进给轴为直线轴。

7) 旋转轴 (Rotary Axis)

旋转运动轨迹的轴，如主轴、回转台回转运动等。

8) 同步轴 (Synchronized Axis)

从起点至任意位置的运动均同步运行的轴。同步轴一般是指一个同步轴组，通常，一个同步轴组中可有两三个轴，其中一个为主同步轴。

9) 通道 (Channel)

在同一时间内，一个通道只能执行一个任务。如果一台机床有多个通道，则在同一时间内，它可以执行多个任务。各个通道既可以单独工作，又可以协同执行任务，它通过通道协同指令来完成。

10) 方式组 (Mode Group)

机床工作方式主要有 JOG 点动、REF 参考点、MDA 手动数据访问、AUTO 自动等，分别用于不同的运行需求。

方式组是指工作方式始终相同的通道组合。一个通道可以是一个方式组，也可以几个通道对应一个方式组。根据机床的应用特性，可以将通道分配到相应的方式组。

11) 参考点 (Reference Point)

机床坐标中各轴的零点位置。所有轴的绝对位置值是以其参考点为参照在相应方向上的偏移值。

12) 机床坐标系 (Machine Reference Frame)

机床坐标系由所有实际存在的机床轴组成。在机床坐标系中定义参考点、换刀点等机床固定点。

坐标系与机床的相互关系取决于机床的类型。按照 ISO841 和我国 JB3051-82 数控标准，数控机床坐标系采用右手笛卡尔坐标系 (图 1)：站到机床面前，伸出右手，中指与主要主轴进刀的方向相对。然后可以得到：

1) 大拇指为方向+X；

2) 食指为方向+Y；

3) 中指为方向+Z。

通常，Z 轴与主轴平行，X 轴垂直于 Z 轴且为水平。机床轴的正向取远离工件的方向。

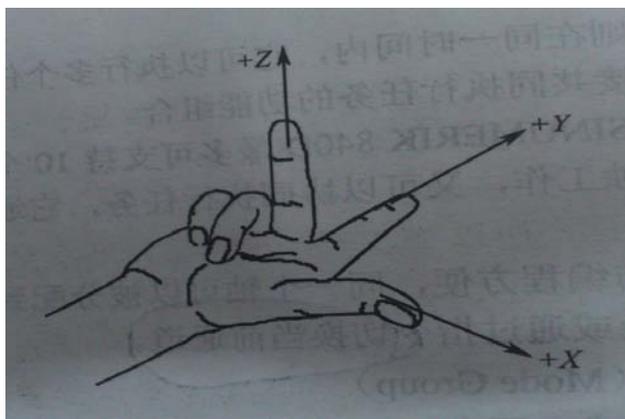


图 1

不同的机床类型其机床轴定义可能会不同，这里给出 2 种常见重型机床的机床坐标系（图 2）。

图 2 中标识的轴向是指刀具在相应方向上的运行方向定义。对于图（a）来说，若是龙门移动，其轴名 X 和方向均如图示；若为工作台移动，则相对工件来说刀具的运动方向相反，其工作台移动的正方向应与图中相反。

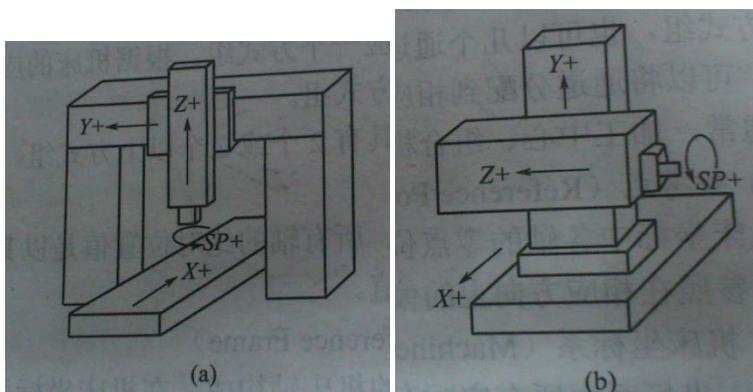


图 2

13) 工件坐标系 (Workpiece Reference Frame)

一台机床安装调试完毕，其机床坐标系就已确定，各点的机床坐标值也是固定的。如果加工时都以机床坐标值编程，则非常方便，且工件加工程序的通用性差，不同的机床加工相同的工件要用不同的工件程序，因此，还需要定义工件坐标系。

根据工件的特点，参照机床坐标系的轴设置，建立一个工件坐标系。工件坐标系始终是直角坐标系，并且与具体的工件相联系。在工件坐标系中确定一个工件零点，据此给出的工件各点几何坐标值来编程，所以 NC 程序中的数据通常是以工件坐标系为基准。

图 3 是一个工件坐标系示例，图中○为工件零点。工件零点相对机床参考点的偏移值称为零点偏置 (Zero Offset Frame)。

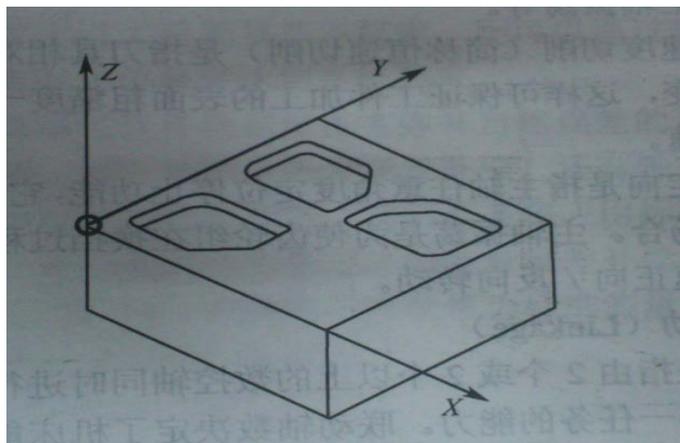


图 3

14) 绝对方式 (Absolute Dimension Mode)

以工件坐标系或机床坐标系中的实际位置值作为编程数据。使用绝对尺寸,所有位置参数均以当前有效的零点为基准。

15) 增量方式 (Incremental Dimension Mode)

在生产过程中经常有一些图纸,其尺寸不是以零点为基准,而是以另外一个工件点为基准。为了避免不必要的尺寸换算,可以使用相对尺寸系统,相对尺寸系统中,输入的尺寸均以在此之前的位置为基准。

数控功能

1) 进给功能 (Feed Function)

通常所说的进给功能包括快速进给、切削进给、点动进给、手动固定增量进给、手动连续进给、进给倍率修调、快移倍率修调、自动加减速等功能。

2) 主轴功能 (Spindle Function)

主轴功能包括恒线速度切削、主轴定向、主轴转速修调、主轴换挡、主轴振荡等。

恒线速度切削(简称恒速切削)是指刀具相对于切削点的速度始终不变,这样可保证工件加工的表面粗糙度一致,其要求主轴自动变速。

主轴定向是指主轴任意角度定位停止功能,它用于主轴换刀、换附件等场合。主轴振荡是为使齿轮组在换挡过程中正常啮合而进行的低速正向/反向转动。

3) 联动 (Linkage)

联动是指由 2 个或 2 个以上的数控轴同时进行成几何关系运动来完成某一任务的能力。联动轴数决定了机床能够完成的轨迹特性。如 2 轴联动只能完成同一平面内的曲线插补,如直线插补,圆弧插补等;三轴联动能够完成螺旋线插补等、球面插补等;5 轴联动机床可以完成许多更复杂的轨迹。

4) 插补 (Interpolation)

插补是在工件轮廓的加工起始点与终止点之间进行“数据密化”并求取中间点的过程。插补功能是指数控机床能够实现的线性加工能力。常用的如直线插补、圆弧插补、螺旋线插补、抛物线插补等。

5) 刀具补偿 (Tool Compensation)

刀具的中心加工点与刀片之间存在距离(刀具半径),且随着刀具加工过程中的磨损或更换新刀等,都会引起其距离的改变;刀尖至接刀杆端部的长度(刀具长度)同样与上述条件相关。要想加工程序不受实际所使用的刀具改变而修改,则必须采用刀具补偿的办法来解决。

刀具补偿是根据刀具外形规格数据来修正进给轴的实际加工位置,使最终刀片轨迹与要求的加工轨迹一致。刀具补偿包括刀具半径补偿和刀具长度补偿。不同的刀具其补偿数据数量不同,刀具加工过程中的磨损可以通过刀具磨损参数来修正。

6) 精度补偿 (Precision Compensation)

机床是一种机电结合的典型设备,其精度取决于多方面因素,单纯依赖传动系统的位置控制很难达到数控精度要求。如果系统传动链定位误差无过大超差且基本稳定,则可以通过软件实现精度补偿,使其实际定位精度达到系统要求。

精度补偿是通过软件数据修正来弥补自然误差的不足。它包括反向间隙补偿、螺距补偿、测量误差补偿,还有轴交叉补偿等。它们都需要应用标准仪器仪表测量自然误差,然后通过软件修正。

控制模式

1) 半闭环控制 (Half-loop Control)

半闭环控制的特点是在轴传动链中的某个部件（如伺服电机、减速箱传动轴、丝杠等）上安装测量装置（如编码器、旋转变压器、感应同步器）作为间接的轴位置反馈检测，通过该反馈号结合传动比计算轴实际位置值。

显而易见，这种控制方法有其局限性，需依赖于机械传动链后半部分的制造、安装调整精度（如丝杠制造精度、丝杠螺母与丝杠的间隙、传动齿轮之间的间隙等）。

由于半闭环控制方式对位置检测器件和安装要求及系统相对简单，另外传动链精度在一定程度上可通过数控系统软件补偿，所以它大量用于轴行程较短（如 2m 以内）、传动惯量较小或定位精度要求相对较低的数控轴控制。

2) 闭环控制 (Loop Control)

闭环控制是采用直接测量装置（如光栅尺、安装在旋转轴末端上的圆光栅等）测量轴实际位置作为轴控制的位置反馈。

它从根本上排除了传动链因制造误差、调整间隙等因素影响轴定位精度，但同样对轴传动系统的要求很高，如传动链的刚度不够、配合间隙过大或因润滑不好等因素引起的导轨摩擦力大等，会造成轴运动爬行或系统运行振荡等现象。

精度指标

1) 精度 (Precision)

精度是指机床能识别，且能精确定位的最小长度单位值。

2) 定位精度 (Position Accuracy)

定位精度是指机床实际位置与指令要求位置值的一致程度，其差值即为误差。误差越大，则定位精度越低。

3) 重复定位精度 (Repeatability)

重复定位精度是指在同样的外界条件下，同样的运行条件下多次测量的定位特性的一致性。若一致性好，则其重复定位精度高。

常用词语

1) 计算机数值控制 (Computerized Numerical Control, CNC) 用计算机控制加工功能，实现数值控制。

2) 机床坐标原点 (Machine Coordinate Origin) 机床坐标系的原点。

3) 工件坐标原点 (Workpiece Coordinate Origin) 工件坐标系原点。

4) 直线插补 (Line Interpolation) 这是一种插补方式，在此方式中，两点间的插补沿着直线的点群来逼近，沿此直线控制刀具的运动。

5) 圆弧插补 (Circular Interpolation) 这是一种插补方式，在此方式中，根据两端点间的插补数字信息，计算出逼近实际圆弧的点群，控制刀具沿这些点运动，加工出圆弧曲线。

6) 顺时针圆弧 (Clockwise Arc) 刀具参考点围绕轨迹中心，按负角度方向旋转所形成的轨迹。方向旋转所形成的轨迹。

7) 逆时针圆弧 (Counterclockwise Arc) 刀具参考点围绕轨迹中心，按正角度方向旋转所形成的轨迹。

8)程序号(Program Number)以号码识别加工程序时, 在每一程序的前端指定的编号

9)程序名(Program Name)以名称识别加工程序时, 为每一程序指定的名称。

10)程序段(Block)程序中为了实现某种操作的一组指令的集合。

11)零件程序(Part Program)在自动加工中, 为了使自动操作有效按某种语言或某种格式书写的顺序指令集。零件程序是写在输入介质上的加工程序, 也可以是为计算机准备的输入, 经处理后得到加工程序。

12)加工程序(Machine Program)在自动加工控制系统中, 按自动控制语言和格式书写的顺序指令集。这些指令记录在适当的输入介质上, 完全能实现直接的操作。

13)程序结束(End of Program)指出工件加工结束的辅助功能。

14)数据结束(End of Data)程序段的所有命令执行完后, 使主轴功能和其他功能(例如冷却功能)均被删除的辅助功能。

15)程序暂停(Program Stop)程序段的所有命令执行完后, 删除主轴功能和其他功能, 并终止其后的数据处理辅助功能。

16)准备功能(Preparatory Function)使机床或控制系统建立加工功能方式的命令。

17)辅助功能(M Function)控制机床或系统的开关功能的一种命令。

18)零点偏置(Zero Offset)数控系统的一种特征.它容许数控测量系统的原点在指定范围内相对于机床零点移动, 但其永久零点则存在数控系统中。

19)刀具偏置(Tool Offset)在一个加工程序的全部或指定部分, 施加于机床坐标轴上的相对位移。该轴的位移方向由偏置值的正负来确定。

20)刀具长度偏置(Tool Length Offset)在刀具长度方向上的偏置。

21)刀具半径偏置(Tool Radius Offset)刀具在两个坐标方向的刀具偏置。

22)刀具半径补偿(Cutter Compensation)垂直于刀具轨迹的位移, 用来修正实际的刀具半径与编程的刀具半径的差异。

23)固定循环(Fixed Cycle , Canned Cycle)预先设定的一些操作命令, 根据这些操作命令使机床坐标轴运动, 主轴工作, 从而完成固定的加工动作。例如, 钻孔、铣削、攻丝以及这些加工的复合动作。

24)子程序(Subprogram)加工程序的一部分, 子程序可由适当的加工控制命令调用而生效。

25)倍率(Override)使操作者在加工期间能够修改速度的编程值(例如, 进给率、主轴转速等)的手工控制功能。

附录六：CNC 通道参数

ID	Structure	Parameter	Functionality
P-CHAN-00001	prog_start.slope.	acceleration	Default slope settings
P-CHAN-00002	vector.	acceleration	Vector dynamic limits
P-CHAN-00003	gruppe[i].	achs_anzahl	Group data
P-CHAN-00004	spindel[i].	autom_range	Spindle speed range
P-CHAN-00005	gruppe[i].	bezeichnung	Group data
P-CHAN-00006	gruppe[i].achse[j].	bezeichnung	Axes data
P-CHAN-00007	spindel[i].	bezeichnung	Spindle data
P-CHAN-00008		cax_face_id	C-axis
P-CHAN-00009		corr_v_trans_jerk	Jerk at tang. block transition
P-CHAN-00010		default_ax_name_of_spindle	C-axis, Tapping
P-CHAN-00011	gruppe[i].achse[j].	default_feed_axis	Feed axes group
P-CHAN-00012	speed_limit_look_ahead.	dist_from_corner	Speed limit Look Ahead
P-CHAN-00013	speed_limit_look_ahead.	dist_to_corner	Speed limit Look Ahead
P-CHAN-00014		einrechnen_mit_t	Tool management
P-CHAN-00015		einzel-schrittmodus	Single step mode
P-CHAN-00016		ext_wzvvorhanden	External tool management
P-CHAN-00017	speed_limit_look_ahead.	f_enable	Speed limit Look Ahead
P-CHAN-00018	speed_limit_look_ahead.	f_time	Speed limit Look Ahead
P-CHAN-00019	syn_chk.	fehler_gesamt	Syntaxcheck
P-CHAN-00020	syn_chk.	fehler_pro_zeile	Syntaxcheck
P-CHAN-00021		create_cont_mask_warnings	Tool radius compensation
P-CHAN-00022	aep.	g_gruppe[i]	Reporting of changes
P-CHAN-00023		grp_anzahl	Group data
P-CHAN-00024	aep.	output_mode	Reporting of changes
P-CHAN-00025		h_default_output_ax_name[i]	Axis specific H-functions
P-CHAN-00026		h_prozess_zeit[i]	Processing time calculation
P-CHAN-00027		h_synch[i]	Techno data
P-CHAN-00028	syn_chk.	interaktiv	Syntaxcheck
P-CHAN-00029		kasto_multi_block	Edge banding
P-CHAN-00030		kasto_residual_path	Edge banding
P-CHAN-00031		kind_of_2nd_ecs_ax	Configuration ECS
P-CHAN-00032		kinematik_id	RTCP
P-CHAN-00033	prog_start.	late_sync_ready	Late Sync at program end
P-CHAN-00034		lin_aufloes	Dimension settings
P-CHAN-00035	gruppe[i].achse[j].	log_achs_nr	Axes data
P-CHAN-00036	spindel[i].	log_achs_nr	Spindle data
P-CHAN-00037	synchro_data.koppel_gruppe[i]. paar[j].	log_achs_nr_master	Synchronous operation
P-CHAN-00038	synchro_data.koppel_gruppe[i]. paar[j].	log_achs_nr_slave	Synchronous operation
P-CHAN-00039		m_default_output_ax_name[i]	Axis specific M-functions
P-CHAN-00040		m_prozess_zeit[i]	Processing time calculation

TwinCAT CNC 系统简明调试教程

P-CHAN-00041		m_synch[i]	Techno data
P-CHAN-00042	spindel[i].	m19_prozess_zeit	Processing time calculation
P-CHAN-00043	spindel[i].	m19_synch	Techno data
P-CHAN-00044	spindel[i].	m3_prozess_zeit	Processing time calculation
P-CHAN-00045	spindel[i].	m3_synch	Techno data
P-CHAN-00046	spindel[i].	m4_prozess_zeit	Processing time calculation
P-CHAN-00047	spindel[i].	m4_synch	Techno data
P-CHAN-00048	spindel[i].	m5_prozess_zeit	Processing time calculation
P-CHAN-00049	spindel[i].	m5_synch	Techno data
P-CHAN-00050		mach_plane_of_2nd_ecs_ax	Configuration ECS
P-CHAN-00051		main_spindle_ax_nr	Spindle configuration
P-CHAN-00052		main_spindle_gear_change	Spindle gear change
P-CHAN-00053		main_spindle_name	Spindle configuration
P-CHAN-00054		mass_einh	Dimension settings
P-CHAN-00055	range_table[j].	max_speed	Spindle speed range
P-CHAN-00056		max_vb_override	Maximum channel override
P-CHAN-00057		messtyp	Default measure type
P-CHAN-00058	range_table[j].	min_speed	Spindle speed range
P-CHAN-00059		mittelpkt_diff	Circle centre point correction
P-CHAN-00060		mittelpkt_faktor	Circle centre point correction
P-CHAN-00061	synchro_data.koppel_gruppe[i]. paar[j].	mode	Synchronous operation
P-CHAN-00062	makro_def[i].	nc_code	Symbolic character strings
P-CHAN-00063	prog_start.g_gruppe[i].	nr	Default G-functions
P-CHAN-00064	prog_start.m_gruppe[i].	nr	Default G-functions
P-CHAN-00065		override_delay	Sampling of override
P-CHAN-00066		override_weight_prog_feed	Mode of override
P-CHAN-00067		p_param_haltend	P-parameter modal
P-CHAN-00068	aep.	p_parameter	Reporting of changes
P-CHAN-00069	spindel[i].	plc_control	Channel specific PLC-spindle
P-CHAN-00070		m_pre_outp[i]	Output definition of M code
P-CHAN-00071	prog_start.slope.	profile	Default slope settings
P-CHAN-00072	tool. (Old syntax: werkzeug.)	prozess_zeit	Processing time calculation
P-CHAN-00073	prog_start.slope.	ramp_time	Default slope settings
P-CHAN-00074	spindel[i].	range_way	Spindle speed range
P-CHAN-00075		reset_no_axis_to_axv	Axis return after reset
P-CHAN-00076		tool_life_to_wzv	Tool life data to external tool management
P-CHAN-00077		rpf_prozess_zeit	Processing time calculation
P-CHAN-00078		rund_aufloes	Dimension settings
P-CHAN-00079		rund_mass_einh	Dimension settings
P-CHAN-00080	spindel[i].	s_prozess_zeit	Processing time calculation
P-CHAN-00081	spindel[i].	s_synch	Techno data
P-CHAN-00082		spdl_anzahl	Spindle configuration
P-CHAN-00083		spind_aufloes	Dimension settings
P-CHAN-00084	gruppe[i].	sub_call	Subcall
P-CHAN-00085	makro_def[i].	symbol	Symbolic character strings

P-CHAN-00086	tool. (Old syntax: werkzeug.)	synch	Techno data
P-CHAN-00087		t_info_to_wzv	External tool management
P-CHAN-00088	aep.	v_eigendef	Reporting of changes
P-CHAN-00089	speed_limit_look_ahead.	v_limit	Speed limit Look Ahead
P-CHAN-00090	vector.	velocity	Vector dynamic limits
P-CHAN-00091		versch_im_durchm	Turning
P-CHAN-00092		wrk_im_kanal_vorhanden	Channel configuration
P-CHAN-00093	aep.	wz_daten	Reporting of changes
P-CHAN-00094	kinematik[i].	param[i] (from V2.10.1501)	RTCP
		wz_kopf_ersatz[i] (to V260)	
P-CHAN-00095		log_number_tracking_axis	Axis tracking
P-CHAN-00096		feed_to_weakest_axis	Selection of default feed axis
P-CHAN-00097		use_drive_curr_limit	Deceleration on feedhold
P-CHAN-00098		spindle_m_fct_free	Meaning of M-Functions
P-CHAN-00099	prog_start.	feedrate	Default feedrate
P-CHAN-00100		move_tool_offsets_directly	Effectiveness of tool compensation data
P-CHAN-00101		auto_align_tracking_axis	Axis tracking
P-CHAN-00102		plc_command_rapid_feed	Mode of PLC command feed
P-CHAN-00103		d_clear_to_wzv	External tool management
P-CHAN-00104	synchro_data.	restore_coupling_after_reset	Synchronous operation
P-CHAN-00105	synchro_data.	preserve_coupling_after_prog_end	Synchronous operation
P-CHAN-00106		t_with_implicit_flush	External tool management
P-CHAN-00107		h_pre_outp[i]	Output definition of H-code
P-CHAN-00108	prog_start.	feedrate_factor	Default unit of feedrate
P-CHAN-00109		fast_tracking_transition	Axis tracking
P-CHAN-00110		check_jerk_on_poly_path	Consider jerk within polynom
P-CHAN-00111		time_override_weight_dwll_time	Time override
P-CHAN-00112		ori_rotation_angle	Kinematic transformation
P-CHAN-00113		path_reject_std_manual_mode	Manual operation mode
P-CHAN-00114		rel_offset_limits_std_manual_mode	Manual operation mode
P-CHAN-00115		release_non_channel_axes	Axis return after reset
P-CHAN-00116		multi_dimension_in_block	Multiple programming of dimensions in NC block
P-CHAN-00117		mode_trans_jerk	Jerk limiting
P-CHAN-00118		m6_prog_file	Sub program call with M6
P-CHAN-00119		start_init_prog_file	Sub program call at program start
P-CHAN-00120		soft_limit_tolerance	Software limit monitoring
P-CHAN-00121		simu_output_wcs	Switching of display data
P-CHAN-00122	hsc.bspline.	path_deviation	HSC BSpline initialization
P-CHAN-00123	hsc.bspline.	track_deviation	HSC BSpline initialization
P-CHAN-00124	hsc.bspline.	max_path_length	HSC BSpline initialization

TwinCAT CNC 系统简明调试教程

P-CHAN-00125	hsc.bspline.	max_angle	HSC BSpline initialization
P-CHAN-00126	hsc.bspline.	merge_retry	HSC BSpline initialization
P-CHAN-00127	hsc.bspline.	merge_window	HSC BSpline initialization
P-CHAN-00128	hsc.bspline.	auto_off_g00	HSC BSpline initialization
P-CHAN-00129	hsc.bspline.	auto_off_path	HSC BSpline initialization
P-CHAN-00130	hsc.bspline.	auto_off_track	HSC BSpline initialization
P-CHAN-00131	hsc.bspline.	limit_corner_dist	HSC BSpline initialization
P-CHAN-00132	hsc.gen.	linear_segmentation	HSC general initialization
P-CHAN-00133	hsc.gen.	linear_center_point	HSC general initialization
P-CHAN-00134	hsc.gen.	linear_continuous_split	HSC general initialization
P-CHAN-00135	hsc.gen.	linear_split_length	HSC general initialization
P-CHAN-00136	hsc.gen.	circular_secant_error	HSC general initialization
P-CHAN-00137	hsc.gen.	circular_segmentation	HSC general initialization
P-CHAN-00138	hsc.gen.	circular_param	HSC general initialization
P-CHAN-00139	hsc.gen.	filter_deviation	HSC general initialization
P-CHAN-00140	hsc.gen.	min_segment_length	HSC general initialization
P-CHAN-00141	hsc.gen.	jerk_monitoring_mode	HSC general initialization
P-CHAN-00142	hsc.gen.	jerk_weighting_5ax	HSC general initialization
P-CHAN-00143	hsc.gen.	jerk_weighting	HSC general initialization
P-CHAN-00144		suppress_cs_tracking_offset	Axis tracking
P-CHAN-00145		kin_trafo_display	Display TCP in WO-CS
P-CHAN-00146		parameter_change_during_execution	Enabling of list interpretation
P-CHAN-00147		soft_limit_warning_axes	Software limit monitoring
P-CHAN-00148		multi_axes_in_block	Multiple programming of axes in NC block
P-CHAN-00149		kasto_reject_rest_block	Edge banding
P-CHAN-00150		remain_block_number_sub_prog_call	Validity of block number
P-CHAN-00151		auto_enable_kin_trafo	<i>Trafoanwahl bei Progr.start</i>
P-CHAN-00152		auto_disable_kin_trafo	<i>Trafoabwahl bei Progr.ende</i>
P-CHAN-00153		kasto_remain_active	Edge banding, measuring
P-CHAN-00154		ax_exchange_with_implicit_flush	Axis exchange with Flush Wait
P-CHAN-00155	speed_limit_look_ahead.	f_override_weight_v_limit	Speed limit Look Ahead
P-CHAN-00156	tool.	minimum_length	Tool monitoring
P-CHAN-00157	tool.	minimum_radius	Tool monitoring
P-CHAN-00158		streaming_prog_file	Streaming
P-CHAN-00159		modal_i_j_k_for_circle	Modal effect of I/J/K for circles
P-CHAN-00160		g80_prog_file	Sub program call at G80
P-CHAN-00161		g81_prog_file	Sub program call at G81
P-CHAN-00162		g82_prog_file	Sub program call at G82
P-CHAN-00163		g83_prog_file	Sub program call at G83
P-CHAN-00164		g84_prog_file	Sub program call at G84
P-CHAN-00165		g85_prog_file	Sub program call at G85
P-CHAN-00166		g86_prog_file	Sub program call at G86
P-CHAN-00167		g87_prog_file	Sub program call at G87
P-CHAN-00168		g88_prog_file	Sub program call at G88
P-CHAN-00169		g89_prog_file	Sub program call at G89

P-CHAN-00170		override_weight_acc	Mode of override
P-CHAN-00171		max_radius_diff_circle	Circle radius compensation
P-CHAN-00172		max_proz_radius_diff_circle	Circle radius compensation
P-CHAN-00173		suppress_ax_group_in_pos_check	Checking exact stop window
P-CHAN-00174		channel_name	Channel information
P-CHAN-00175		channel_type	Channel information
P-CHAN-00176		meas_error_no_signal	Reaction on missing signal for measurement type 1
P-CHAN-00177		mode	HSC Punkt-Vektordarstellung
P-CHAN-00178		fixed_axis_index	HSC Punkt-Vektordarstellung
P-CHAN-00179		ax_default_config_after_reset	Channel axes after RESET
P-CHAN-00180		suppress_prg_display_level	Display data
P-CHAN-00181		g00_override_mode	Mode of override
P-CHAN-00182	Nur Homag	plcopen_std_unit	Units in PLCopen commands
P-CHAN-00183		simu_ignore_internal_stop_cond	Settings for rapid contour visu.
P-CHAN-00184	111	kin_trafo_display_curr_pos	Switch display TCP in WO-CS
P-CHAN-00185		tracking_axis_rot_wc	Axis tracking
P-CHAN-00186		override_v_handwheel	Override at manual mode
P-CHAN-00187		g800_prog_file[i]	Sub program call at G800 – G819
P-CHAN-00188	tool_ori_cs.	axis	Automatic orientation
P-CHAN-00189	tool_ori_cs. 112	mode	Automatic orientation
P-CHAN-00190		dynamic_weighting_active	Path dependend dynamic
P-CHAN-00191	dynamic_weighting[i].	path_limit	Path dependend dynamic
P-CHAN-00192	dynamic_weighting[i].	velocity_fact	Path dependend dynamic
P-CHAN-00193	dynamic_weighting[i].	acceleration_fact	Path dependend dynamic
P-CHAN-00194	dynamic_weighting[i].	ramp_time_fact	Path dependend dynamic
P-CHAN-00195	man_mode.vector_limit. 154	v_max_pos	Dynamikdaten kart. Handbetrieb
P-CHAN-00196	man_mode.vector_limit. 154	a_max_pos	Dynamikdaten kart. Handbetrieb
P-CHAN-00197	man_mode.vector_limit. 154	tr_pos	Dynamikdaten kart. Handbetrieb
P-CHAN-00198	man_mode.vector_limit. 155	v_max_ori	Dynamikdaten kart. Handbetrieb
P-CHAN-00199	man_mode.vector_limit. 155	a_max_ori	Dynamikdaten kart. Handbetrieb
P-CHAN-00200	man_mode.vector_limit. 155	tr_ori	Dynamikdaten kart. Handbetrieb
P-CHAN-00201	hsc.filter.	auto_enable	HSC filter parameter
P-CHAN-00202	hsc.filter.	order_cart	HSC filter parameter
P-CHAN-00203	hsc.filter.	order_track	HSC filter parameter
P-CHAN-00204	hsc.filter.		HSC filter parameter
P-CHAN-00205	hsc.filter.		HSC filter parameter
P-CHAN-00206	hsc.filter.		HSC filter parameter
P-CHAN-00207	hsc.filter.		HSC filter parameter
P-CHAN-00208	vector.	deceleration	Vector dynamic limits
P-CHAN-00209		m_h_pre_outp_time_calc_mode	Calculating pre-output time
P-CHAN-00210		cycle_changes_modal	Cycle programming
P-CHAN-00211		suppress_cycle_logging	Cycle programming
P-CHAN-00212		m_h_pre_outp_calc_value_to_go	Calculating residual path/time

P-CHAN-00213		gap_in_trafo_axis_sequence	Transformations
P-CHAN-00214		meas_deceleration_mode	Deceleration on meas. signal
P-CHAN-00215		consider_rot_tracking_offset	Rotate tracking axis
P-CHAN-00216		max_nc_blocks_ahead	Decoder block ahead limiting
P-CHAN-00217	hsc.gen.	on_delay_output_one_block	HSC general initialization
P-CHAN-00218		independent_axis_error_stop	Behaviour on axis error
P-CHAN-00219	<i>vorläufig</i>	<i>ipo_start_wait_cycles (Ba , Bystronic)</i>	
P-CHAN-00220	edge_machining.	f_enable	Edge machining function
P-CHAN-00221	edge_machining.	angle_limit	Edge machining function
P-CHAN-00222	edge_machining.	pre_dist	Edge machining function
P-CHAN-00223	edge_machining.	pre_feed	Edge machining function
P-CHAN-00224	edge_machining.	wait_time	Edge machining function
P-CHAN-00225	edge_machining.	post_dist	Edge machining function
P-CHAN-00226	edge_machining.	post_feed	Edge machining function
P-CHAN-00227		51 kasto_suppress_path_check	Edge banding
P-CHAN-00228	<i>Roland</i>	permitted_trafo_deviation_limit	Transformationen
P-CHAN-00229	<i>reserviert</i>		
P-CHAN-00230		curve_dynamic_weighting_active	Radiusabhängige Dynamik
P-CHAN-00231	curve_dynamic_weighting[i].	radius_limit	Radiusabhängige Dynamik
P-CHAN-00232	curve_dynamic_weighting[i].	velocity_fact	Radiusabhängige Dynamik
P-CHAN-00233	curve_dynamic_weighting[i].	acceleration_fact	Radiusabhängige Dynamik
P-CHAN-00234	curve_dynamic_weighting[i].	ramp_time_fact	Radiusabhängige Dynamik

附录七：CNC 轴参数列表

ID	Structure	Parameter	for drive type								axis types			Used in functionality/ BF
			Simu	Conv	SERC	Term	Light	Profi	CAN-open	DSE	L	R	S	
P-AXIS-00001	getriebe[i].slope_profil.	a_beschl	X	X	X	X	X	X	X		X	X	X	Dynamic value for nonlinear slope
P-AXIS-00002	getriebe[i].slope_profil.	a_brems	X	X	X	X	X	X	X		X	X	X	Dynamic value for nonlinear slope
P-AXIS-00003	getriebe[i].dynamik.	a_emergency	X	X	X	X	X	X	X		X	X	X	Emergency stop
P-AXIS-00004	getriebe[i].slope_profil.	a_grenz	X	X	X	X	X	X	X		X	X	X	Dynamic value for nonlinear slope
P-AXIS-00005	getriebe[i].lslope_profile.	a_grenz_stufe_1	X	X	X	X	X	X	X		X	X	X	Dynamic value for linear slope
P-AXIS-00006	getriebe[i].lslope_profile.	a_grenz_stufe_2	X	X	X	X	X	X	X		X	X	X	Dynamic value for linear slope
P-AXIS-00007	getriebe[i].beschlnlinie.	a_konst	X	X	X	X	X	X	X				X	Spindle interpolator
P-AXIS-00008	getriebe[i].dynamik.	a_max	X	X	X	X	X	X	X		X	X	X	Dynamic value for lin. + nonlin. slope
P-AXIS-00009	handbetrieb.hb.	a_max	X	X	X	X	X	X	X		X	X		Manual mode
P-AXIS-00010	getriebe[i].beschlnlinie.	a_min	X	X	X	X	X	X	X				X	Spindle interpolator

P-AXIS-00011	getriebe[i].Islope_prof il.	a_stufe_1	X	X	X	X	X	X	X	X	X	X	X	Dynamic value for linear slope
P-AXIS-00012	getriebe[i].Islope_prof il.	a_stufe_2	X	X	X	X	X	X	X	X	X	X	X	Dynamic value for linear slope
P-AXIS-00013	getriebe[i].dynamik.	a_trans_weight	X	X	X	X	X	X	X	X	X	X	X	Dynamic value for nonlinear slope
P-AXIS-00014	kenngr.	abs_pos_gueltig	X		X						X	X	X	Absolute measurement system
P-AXIS-00015	kenngr.	achs_mode	X	X	X	X	X	X	X	X	X	X	X	Axis mode
P-AXIS-00016	kopf.	achs_nr	X	X	X	X	X	X	X	X	X	X	X	Axis number
P-AXIS-00017	<i>getriebe[i].</i>	<i>achs_position[j]</i>	X	X	X	X	X	X	X	X			X	<i>Single axis interpolator</i>
P-AXIS-00018	kenngr.	achs_typ	X	X	X	X	X	X	X	X	X	X	X	Axis type
P-AXIS-00019	antr.sercos.	antr_adr			X						X	X	X	Configuration SERCOS
P-AXIS-00020	kenngr.	antr_typ	X	X	X	X	X	X	X	X	X	X	X	Drive type
P-AXIS-00021	lr_param.	anwahl_losekomp	X	X	X	X	X	X	X	X	X	X	X	Compensation
P-AXIS-00022	<i>lr_param.</i>	<i>anz_driftwerte</i>												<i>Position controler</i>
P-AXIS-00023	kenngr.	anzeige	X	X	X	X	X	X	X	X	X	X	X	Position controler
P-AXIS-00024	getriebe[i].Islope_prof il	a_feedh	X	X	X	X	X	X	X	X	X	X		Dynamic values for linear slope
P-AXIS-00025	handbetrieb.hr.	auf1[i]	X	X	X	X	X	X	X	X	X	X		Manual mode
P-AXIS-00026	getriebe[i].beschl_ken nlinie.	b1	X	X	X	X	X	X	X	X			X	Spindle interpolator
P-AXIS-00027	getriebe[i].beschl_ken nlinie.	b2	X	X	X	X	X	X	X	X			X	Spindle interpolator
P-AXIS-00028	getriebe[i].beschl_ken nlinie.	b3	X	X	X	X	X	X	X	X			X	Spindle interpolator

P-AXIS-00029	<i>bahnrg.</i>	<i>bahnregelung</i>												<i>Path control</i>
P-AXIS-00030	getriebe[i].	vb_max_red_zone	X	X	X	X	X	X	X	X		X	X	Velocity in security zone
P-AXIS-00031	kenngr.	beweg_richt	X	X	X	X	X	X	X	X			X	Rotary axis
P-AXIS-00032	<i>bahnrg.</i>	<i>br_i_faktor</i>												<i>Path control</i>
P-AXIS-00033	<i>bahnrg.</i>	<i>br_i_nenner</i>												<i>Path control</i>
P-AXIS-00034	<i>bahnrg.</i>	<i>br_p_faktor</i>												<i>Path control</i>
P-AXIS-00035	<i>bahnrg.</i>	<i>br_p_nenner</i>												<i>Path control</i>
P-AXIS-00036	lr_hw[i].	cam_direct_access		X		X	X	X				X	X	X
P-AXIS-00037	lr_hw[i].	cam_hw_id_string		X		X	X	X				X	X	X
P-AXIS-00038	lr_hw[i].	cam_level		X		X	X	X				X	X	X
P-AXIS-00039	lr_hw[i].	cam_mask		X		X	X	X				X	X	X
P-AXIS-00040	kopf.	clone_of	X	X	X	X	X	X	X			X	X	X
P-AXIS-00041	lr_hw[i].	cntr_channel		X		X	X	X				X	X	X
P-AXIS-00042	lr_hw[i].	cntr_hw_id_string		X		X	X	X				X	X	X
P-AXIS-00043	kenngr.	coll_check_ax_nr	X	X	X	X	X	X	X			X		Collision detection
P-AXIS-00044	kenngr.	coll_decelerate_chan			X							X		Collision detection
P-AXIS-00045	kenngr.	coll_offset	X	X	X	X	X	X	X			X		Collision detection
P-AXIS-00046	handbetrieb.default.	control_element	X	X	X	X	X	X	X			X	X	Manual mode
P-AXIS-00047	lr_param.	crosscomp	X	X	X	X	X	X	X			X		Compensation
P-AXIS-00048	lr_hw[i].	da_channel		X		X	X	X				X	X	X
P-AXIS-00049	lr_hw[i].	da_hw_id_string		X		X	X	X				X	X	X
P-AXIS-00050	antr.simu	daempfung_n	X									X	X	X
P-AXIS-00051	antr.simu	daempfung_z	X									X	X	X

P-AXIS-00052	kenngr.	def_cax_gear_st	X	X	X	X	X	X	X		X	X	C-Achse
P-AXIS-00053	getriebe[i].slope_profi l.	a_feedh	X	X	X	X	X	X	X		X	X	Dynamic value for nonlinear slope
P-AXIS-00054	getriebe[i].	default_lr_hw_nbr	X	X	X	X	X	X	X		X	X	Gear shift
P-AXIS-00055	<i>vorsteuer.</i>	<i>delay_time</i>											<i>Adaptive feedforward control</i>
P-AXIS-00056	kenngr.	diff_pos_tracking											Tracking operation
P-AXIS-00057	lr_hw[i].	drift_wert		X		X	X	X			X	X	Compensation
P-AXIS-00058	kenngr.	durchm_prog_abs	X	X	X	X	X	X	X		X		Turning
P-AXIS-00059	kenngr.	durchm_prog_rel	X	X	X	X	X	X	X		X		Turning
P-AXIS-00060	kenngr.	echtzeit_bit_nr			X						X	X	Measurement with SERCOS
P-AXIS-00061	antr.simu	eigenfrequenz_n	X								X	X	Axis simulation
P-AXIS-00062	antr.simu	eigenfrequenz_z	X								X	X	Axis simulation
P-AXIS-00063	antr.sercos.	eval_calc_slot			X						X	X	Configuration SERCOS
P-AXIS-00064	kenngr.	fast_from_cam	X	X	X	X	X	X	X		X	X	Homing
P-AXIS-00065	antr.profibus.	feinauflösung						X			X	X	Feedforward control
P-AXIS-00066	<i>kenngr.</i>	<i>feininterpolation</i>											<i>Fine interpolation</i>
P-AXIS-00067	filter[i].	fg_f0	X	X	X	X	X	X	X		X		Position setpoint filter
P-AXIS-00068	antr.sercos.ident[i].	file			X						X	X	Configuration SERCOS
P-AXIS-00069	handbetrieb.hr.	filter_zeit	X	X	X	X	X	X	X		X	X	Manual mode
P-AXIS-00070	kenngr.	gantry_ax_nr	X	X	X	X	X	X	X		X	X	Gantry operation
P-AXIS-00071	kenngr.	gantry_max_diff_reset_lock	X	X	X	X	X	X	X		X	X	Gantry

		ed											operation
P-AXIS-00072	kenngr.	gantry_max_diff_resetable	X	X	X	X	X	X	X	X	X	X	Gantry operation
P-AXIS-00073	kenngr.	gantry_offset	X	X	X	X	X	X	X	X	X	X	Gantry operation
P-AXIS-00074	kenngr.	gantry_slave_no_homing	X	X	X	X	X	X	X	X	X	X	Gantry operation
P-AXIS-00075	kenngr.	gantry_vb_korr	X	X	X	X	X	X	X	X	X	X	Gantry operation
P-AXIS-00076	handbetrieb.jog.	geschw[i]	X	X	X	X	X	X	X	X	X	X	Manual mode
P-AXIS-00077	handbetrieb.tipp.	geschw[i]	X	X	X	X	X	X	X	X	X	X	Manual mode
P-AXIS-00078	getriebe[i].	getr_schalt_pos	X	X		X	X	X	X	X	X	X	Spindle
P-AXIS-00079	kenngr.	getriebe_stufe	X	X	X	X	X	X	X	X	X	X	Gear step
P-AXIS-00080	filter[i].	guete	X	X	X	X	X	X	X	X	X		Position setpoint filter
P-AXIS-00081	getriebe[i].slope_profile.	tr_feedh	X	X	X	X	X	X	X	X	X	X	Dynamic value for nonlinear slope
P-AXIS-00082	handbetrieb.ipos.	hb_proz_a_max	X	X	X	X	X	X	X	X	X	X	Manual mode
P-AXIS-00083	handbetrieb.ipos.	hb_proz_v_max	X	X	X	X	X	X	X	X	X	X	Manual mode
P-AXIS-00084	kenngr.	homing_without_zero_pulse	X	X	X	X	X	X	X	X	X	X	Homing
P-AXIS-00085	kenngr.	red_speed_zone_pos	X	X	X	X	X	X	X	X	X	X	Monitoring of security zone 1
P-AXIS-00086	kenngr.	hub_messtaster	X	X	X	X	X	X	X	X	X	X	Measurement
P-AXIS-00087	antr.sercos.at[i].	ident_len			X						X	X	Configuration SERCOS
P-AXIS-00088	antr.sercos.mdt[i].	ident_len			X						X	X	Configuration SERCOS
P-AXIS-00089	antr.sercos.at[i].	ident_nr			X						X	X	Configuration SERCOS
P-AXIS-00090	antr.sercos.mdt[i].	ident_nr			X						X	X	Configuration

														SERCOS	
P-AXIS-00091	kenngr.	in_add_interface	X	X	X	X	X	X	X	X		X	X	X	Position controller
P-AXIS-00092	vorsteuer.	incr_per_rev							X			X	X	X	Feedforward control
P-AXIS-00093	kenngr.	red_speed_zone_neg	X	X	X	X	X	X	X	X		X	X		Monitoring of security zone 1
P-AXIS-00094	handbetrieb.ipo.	ipo_proz_a_max	X	X	X	X	X	X	X	X		X	X		Manual mode
P-AXIS-00095	handbetrieb.ipo.	ipo_proz_v_max	X	X	X	X	X	X	X	X		X	X		Manual mode
P-AXIS-00096	antr.simu	ist_gleich_soll	X									X	X	X	Axis simulation
P-AXIS-00097	kenngr.	red_speed_zone_2_pos	X	X	X	X	X	X	X	X		X	X		Monitoring of security zone 2
P-AXIS-00098	kenngr.	kasto_achse	X	X	X	X	X	X	X	X		X	X		Edge bending
P-AXIS-00099	getriebe[i].	kv	X	X		X	X	X	X	X		X	X	X	Position control
P-AXIS-00100	antr.sercos.ident[i].	laenge			X							X	X	X	Configuration SERCOS
P-AXIS-00101	kopf.	link_to	X	X	X	X	X	X	X	X		X	X	X	Create multiple instances
P-AXIS-00102	antr.sercos.ident[i].	liste			X							X	X	X	Configuration SERCOS
P-AXIS-00103	getriebe[i].	lose	X	X	X	X	X	X	X	X		X	X	X	Compensation
P-AXIS-00104	<i>lr_param.</i>	<i>lr_ueberwach</i>													<i>Position controle</i>
P-AXIS-00105	kenngr.	red_speed_zone_2_neg	X	X	X	X	X	X	X	X		X	X		Monitoring of security zone 2
P-AXIS-00106	lr_hw[i].	mask_mess_1			X							X	X		Measurement with Sercos
P-AXIS-00107	<i>kenngr.</i>	<i>max_a_override</i>													<i>Acceleration override</i>
P-AXIS-00108	kenngr.	max_diff_soll_ist	X	X	X	X	X	X	X	X		X	X	X	Enabling of drive controller
P-AXIS-00109	kenngr.	max_vb_override	X	X	X	X	X	X	X	X		X	X	X	Velocity override
P-AXIS-00110	<i>kopf.</i>	<i>mds_ident</i>													<i>Interpretation of</i>

														Binary list
P-AXIS-00111	antr.sercos.	mdt_offset			X						X	X	X	Configuration SERCOS
P-AXIS-00112	meas_simu.	meas_simu_mode	X								X	X		Measurement
P-AXIS-00113	kenngr.	mess_neg_flanke	X	X	X	X	X	X	X		X	X		Measurement
P-AXIS-00114	kenngr.	mess_offset	X	X	X	X	X	X	X		X	X		Measurement
P-AXIS-00115	kenngr.	mess_signal_achs_steuer	X	X	X	X	X	X	X		X	X		Measurement
P-AXIS-00116	kenngr.	mess_signal_sercos			X						X	X		Measurement with SERCOS
P-AXIS-00117	kenngr.	mess_signal_taster	X	X	X	X	X	X	X		X	X		Measurement
P-AXIS-00118	kenngr.	messachse	X	X	X	X	X	X	X		X	X		Measurement
P-AXIS-00119	antr.sercos.ident[i].	mod			X						X	X	X	Configuration SERCOS
P-AXIS-00120	lr_param.	mod_komp	X	X	X	X	X	X	X			X	X	Compensation
P-AXIS-00121	kenngr.	mod_rechnng												Modulo calculation
P-AXIS-00122	antr.	mode_act_pos			X						X	X	X	Normalization of actual position
P-AXIS-00123	antr.	mode_cmd_pos			X						X	X	X	Normalization of setpoint position
P-AXIS-00124	getriebe[i].	modulo_fehler	X	X		X	X	X	X			X	X	Compensation
P-AXIS-00125	getriebe[i].	modulo_umdreh	X	X		X	X	X				X	X	Compensation
P-AXIS-00126	getriebe[i].	moduloo	X	X	X	X	X	X	X			X	X	Spindle, rot. axes
P-AXIS-00127	getriebe[i].	modulou	X	X	X	X	X	X	X			X	X	Spindle, rot. axes
P-AXIS-00128	getriebe[i].	multi_gain_n	X	X		X	X	X			X	X	X	Position control
P-AXIS-00129	getriebe[i].	multi_gain_z	X	X		X	X	X			X	X	X	Position control
P-AXIS-00130	getriebe[i].beschl_ken nlinie.	n_grenz	X	X	X	X	X	X	X				X	Spindle
P-AXIS-00131	antr.sercos.at[i].	nc_ref			X						X	X	X	Configuration SERCOS

P-AXIS-00132	antr.sercos.mdt[i].	nc_ref			X						X	X	X	Configuration SERCOS
P-AXIS-00133	kenngr.	nocken_laenge												Cam length
P-AXIS-00134	antr.sercos.ident[i].	nr			X						X	X	X	Configuration SERCOS
P-AXIS-00135	getriebe[i].	nummer	X	X	X	X	X	X	X		X	X	X	Gear change
P-AXIS-00136	lr_hw[i].	nummer	X	X		X	X	X			X	X	X	Position controler
P-AXIS-00137	handbetrieb.	offsetgrenze_neg	X	X	X	X	X	X	X		X	X		Manual mode
P-AXIS-00138	handbetrieb.	offsetgrenze_pos	X	X	X	X	X	X	X		X	X		Manual mode
P-AXIS-00139	handbetrieb.default.	operation_mode	X	X	X	X	X	X	X		X	X		Manual mode
P-AXIS-00140	filter[i].	order	X	X	X	X	X	X	X		X			Position setpoint filter
P-AXIS-00141	kenngr.	out_add_interface	X	X	X	X	X	X	X		X	X	X	Position controler
P-AXIS-00142	kenngr.	paar_achse												Rotation-/swivel joint
P-AXIS-00143	kenngr.	paar_ax_nr												Rotation-/swivel joint
P-AXIS-00144	kenngr.	paar_ref_richt												Rotation-/swivel joint
P-AXIS-00145	meas_simu.	parameter1	X								X	X		Measurement
P-AXIS-00146	meas_simu	parameter2												Measuerement simulation
P-AXIS-00147	meas_simu	parameter3												Measuerement simulation
P-AXIS-00148	meas_simu	parameter4												Measuerement simulation
P-AXIS-00149	meas_simu	parameter5												Measuerement simulation
P-AXIS-00150	antr.sercos.ident[i].	phase			X						X	X	X	Configuration SERCOS

P-AXIS-00151	getriebe[i].	pos_einschw_zeit	X	X	X	X	X	X	X	X	X	X	X	Monitoring
P-AXIS-00152	getriebe[i].	pos_refpkt	X	X		X	X	X			X	X	X	Homing
P-AXIS-00153	filter[i].	prototype	X	X	X	X	X	X	X	X	X			Position setpoint filter
P-AXIS-00154	getriebe[i].dynamik.	r_trans_weight	X	X	X	X	X	X	X	X	X	X	X	Dynamic value for nonlinear slope
P-AXIS-00155	getriebe[i].	rapid_speed_red	X	X	X	X	X	X	X	X	X	X	X	Monitoring security zone
P-AXIS-00156	kenngr.	ref_ohne_nocken		X		X	X	X			X	X	X	Homing
P-AXIS-00157	kenngr.	ref_ohne_rev		X		X	X	X			X	X	X	Homing
P-AXIS-00158	kenngr.	ref_richt	X	X		X	X	X			X	X	X	Homing
P-AXIS-00159	getriebe[i].	reverse	X	X		X	X	X					X	Spindle
P-AXIS-00160	antr.sercos.	ring_nr			X						X	X	X	Configuration SERCOS
P-AXIS-00161	antr.simu	rpf_weg_bis_nip	X								X	X	X	Axis simulation
P-AXIS-00162	antr.profibus.	s_ls_limit						X			X	X	X	Monitoring, Profidrive
P-AXIS-00163		regel_typ												Position controller
P-AXIS-00164	filter[i].	share_percent	X	X	X	X	X	X	X	X	X			Position setpoint filter
P-AXIS-00165	vorsteuer.	shift_time						X			X	X	X	Feedforward control, Profidrive
P-AXIS-00166	getriebe[i].	slep_abw	X	X		X	X	X	X	X	X	X	X	Dynamical monitoring of following error
P-AXIS-00167	getriebe[i].	slep_dyn	X	X		X	X	X	X	X	X	X	X	Dynamical monitoring of following error
P-AXIS-00168	getriebe[i].	slep_max	X	X		X	X	X	X	X	X	X	X	Dynamical

														monitoring of following error
P-AXIS-00169	getriebe[i].	slep_min	X	X		X	X	X	X		X	X	X	Dynamical monitoring of following error
P-AXIS-00170	getriebe[i].	slep_time_const						X	X		X	X		Dynamical monitoring of following error
P-AXIS-00171	<i>lr_param.</i>	<i>slep_toleranz</i>												<i>Position controler</i>
P-AXIS-00172	getriebe[i].	slep_ueberw_typ	X	X		X	X	X	X		X	X	X	Dynamical monitoring of following error
P-AXIS-00173	antr.sercos.times.	slkn			X						X	X	X	Configuration SERCOS
P-AXIS-00174	lr_param.	crosscomp2	X	X	X	X	X	X	X		X			Compensation
P-AXIS-00175	lr_param.	ssfk	X	X	X	X	X	X	X		X	X		Compensation
P-AXIS-00176	lr_param.	suppress_pos_lag_error						X	X		X	X		Dynamical monitoring of following error
P-AXIS-00177	kenngr.	swe_neg	X	X	X	X	X	X	X		X	X		Software limit monitoring
P-AXIS-00178	kenngr.	swe_pos	X	X	X	X	X	X	X		X	X		Software limit monitoring
P-AXIS-00179	kenngr.	swe_toleranz	X	X		X	X	X			X	X		Software limit monitoring
P-AXIS-00180	antr.sercos.times.	t1			X						X	X	X	Configuration SERCOS
P-AXIS-00181	antr.sercos.times.	t1min			X						X	X	X	Configuration SERCOS
P-AXIS-00182	antr.sercos.times.	t2			X						X	X	X	Configuration SERCOS
P-AXIS-00183	antr.sercos.times.	t3			X						X	X	X	Configuration

														SERCOS
P-AXIS-00184	antr.sercos.times.	t4			X						X	X	X	Configuration SERCOS
P-AXIS-00185	antr.sercos.times.	t4min			X						X	X	X	Configuration SERCOS
P-AXIS-00186	antr.sercos.times.	tatat			X						X	X	X	Configuration SERCOS
P-AXIS-00187	antr.sercos.times.	tatmt			X						X	X	X	Configuration SERCOS
P-AXIS-00188	antr.sercos.	telegramm_typ			X						X	X	X	Configuration SERCOS
P-AXIS-00189	kenngr.	tendenz_pruef	X	X	X	X	X	X			X	X	X	Monitoring
P-AXIS-00190	<i>vorsteuer.</i>	<i>timeconst_cmd_filter</i>												<i>Adaptive feedforward control</i>
P-AXIS-00191	antr.	nbr_delay_cycles			X				X	X	X	X	X	Configuration digital drives
P-AXIS-00192	antr.sercos.times.	tmtsg			X						X	X	X	Configuration SERCOS
P-AXIS-00193	antr.sercos.times.	tmtsy			X						X	X	X	Configuration SERCOS
P-AXIS-00194	antr.simu	totzeit	X								X	X	X	Axis simulation
P-AXIS-00195	getriebe[i].slope_profi l.	tr_beschl_ab	X	X	X	X	X	X	X		X	X	X	Dynamic value for nonlinear slope
P-AXIS-00196	getriebe[i].slope_profi l.	tr_beschl_zu	X	X	X	X	X	X	X		X	X	X	Dynamic value for nonlinear slope
P-AXIS-00197	getriebe[i].slope_profi l.	tr_brems_ab	X	X	X	X	X	X	X		X	X	X	Dynamic value for nonlinear slope
P-AXIS-00198	getriebe[i].slope_profi l.	tr_brems_zu	X	X	X	X	X	X	X		X	X	X	Dynamic value for nonlinear

														slope	
P-AXIS-00199	getriebe[i].dynamik.	tr_geom	X	X	X	X	X	X	X	X		X	X	X	Dynamic value for nonlinear slope
P-AXIS-00200	getriebe[i].slope_profile.	tr_grenz	X	X	X	X	X	X	X	X		X	X	X	Dynamic value for nonlinear slope
P-AXIS-00201	getriebe[i].dynamik.	tr_min	X	X	X	X	X	X	X	X		X	X	X	Dynamic value for nonlinear slope
P-AXIS-00202	getriebe[i].beschl_kennlinie.	typ	X	X	X	X	X	X	X	X				X	Spindle interpolator
P-AXIS-00203	antr.sercos.ident[i].	type			X							X	X	X	Configuration SERCOS
P-AXIS-00204	filter[i].	type	X	X	X	X	X	X	X	X		X			Position setpoint filter
P-AXIS-00205	antr.	v_reso_denom			X							X	X	X	Standardization of velocity
P-AXIS-00206	antr.	v_reso_num			X							X	X	X	Standardization of velocity
P-AXIS-00207	antr.	v_time_base			X							X	X	X	Standardization of velocity
P-AXIS-00208	kenngr.	vb_corr_tracking													Tracking operation
P-AXIS-00209	getriebe[i].	vb_eilgang	X	X	X	X	X	X	X	X		X	X	X	Rapid movement feedrate
P-AXIS-00210	handbetrieb.tipp.	vb_eilgang	X	X	X	X	X	X	X	X		X	X		Manual mode
P-AXIS-00211	getriebe[i].lslope_profile.	vb_grenz_stufe_1_2	X	X	X	X	X	X	X	X		X	X	X	Dynamic value for linear slope
P-AXIS-00212	getriebe[i].dynamik.	vb_max	X	X	X	X	X	X	X	X		X	X	X	Dynamic value for lin. + nonlin.

														slope
P-AXIS-00213	handbetrieb.hb.	vb_max	X	X	X	X	X	X	X	X	X	X	X	Manual mode
P-AXIS-00214	getriebe[i].	vb_max_red	X	X	X	X	X	X	X	X	X	X	X	Monitoring security zone
P-AXIS-00215	kenngr.	vb_messen	X	X	X	X	X	X	X	X	X	X	X	Measurement
P-AXIS-00216	getriebe[i].	vb_min_null	X	X	X	X	X	X	X	X			X	Spindle
P-AXIS-00217	kenngr.	vb_prozent	X	X	X	X	X	X	X	X			X	Spindle
P-AXIS-00218	getriebe[i].	vb_reflow	X	X		X	X	X			X	X	X	Homing
P-AXIS-00219	getriebe[i].	vb_refmax		X		X	X	X			X	X	X	Homing
P-AXIS-00220	getriebe[i].	vb_regelgrenze	X	X		X	X	X					X	Spindle
P-AXIS-00221	getriebe[i].Islope_prof il.	vb_stufe_1_2	X	X	X	X	X	X	X	X	X	X	X	Dynamic value for linear slope
P-AXIS-00222	kenngr.	virtuel_rad												Virtual radius
P-AXIS-00223	vorsteuer.	vorsteuerung	X	X	X	X	X	X	X	X	X	X	X	Feedforward control
P-AXIS-00224	kenngr.	vorz_richtung	X	X	X	X	X	X	X	X		X	X	Rotatary axis
P-AXIS-00225	vorsteuer.	vs_a_faktor	X	X	X	X	X	X	X	X	X	X	X	Feedforward control
P-AXIS-00226	vorsteuer.	vs_a_nenner	X	X	X	X	X	X	X	X	X	X	X	Feedforward control
P-AXIS-00227	vorsteuer.	vs_max_sprung												Feedforward control
P-AXIS-00228	vorsteuer.	vs_v_faktor	X	X	X	X	X	X	X	X	X	X	X	Feedforward control
P-AXIS-00229	vorsteuer.	vs_v_nenner	X	X	X	X	X	X	X	X	X	X	X	Feedforward control
P-AXIS-00230	lr_hw[i].	vz_istw	X	X		X	X	X			X	X	X	Position controler
P-AXIS-00231	lr_hw[i].	vz_stellgr	X	X		X	X	X			X	X	X	Position controler
P-AXIS-00232	handbetrieb.jog.	weg[i]	X	X	X	X	X	X	X	X	X	X		Manual mode
P-AXIS-00233	getriebe[i].	wegaufn	X	X	X	X	X	X	X	X	X	X	X	Position

														controller	
P-AXIS-00234	getriebe[i].	wegaufz	X	X	X	X	X	X	X	X		X	X	X	Position controller
P-AXIS-00235	antr.sercos.ident[.].	wert			X							X	X	X	Configuration SERCOS
P-AXIS-00236	getriebe[i].	window	X	X	X	X	X	X	X	X		X	X	X	Monitoring
P-AXIS-00237	getriebe[i].	wsi_meldung	X	X	X	X	X	X	X	X		X	X	X	Monitoring
P-AXIS-00238	antr.simu	zeitkonstante_n	X									X	X	X	Axis simulation
P-AXIS-00239	antr.simu	zeitkonstante_z	X									X	X	X	Axis simulation
P-AXIS-00240	getriebe[i].beschl_ken nlinie.	a_max	X	X	X	X	X	X	X	X				X	Spindle
P-AXIS-00241	getriebe[i].beschl_ken nlinie.	n1	X	X	X	X	X	X	X	X				X	Spindle
P-AXIS-00242	getriebe[i].beschl_ken nlinie.	n2	X	X	X	X	X	X	X	X				X	Spindle
P-AXIS-00243	lr_param.	n_backlash_cyc	X	X	X	X	X	X	X	X		X	X	X	Compensation
P-AXIS-00244	getriebe[i].dyn_kv.	dyn_kv_active	X	X		X	X	X				X	X	X	Position controller
P-AXIS-00245	getriebe[i].dyn_kv.	kv1	X	X		X	X	X				X	X	X	Position controller
P-AXIS-00246	getriebe[i].dyn_kv.	v1	X	X		X	X	X				X	X	X	Position controller
P-AXIS-00247	getriebe[i].dyn_kv.	kv2	X	X		X	X	X				X	X	X	Position controller
P-AXIS-00248	getriebe[i].dyn_kv.	v2	X	X		X	X	X				X	X	X	Position controller
P-AXIS-00249	kenngr.	gantry_diff_check_without_ homing	X	X	X	X	X	X	X	X		X	X		Gantry operation
P-AXIS-00250	kenngr.	configure_sai	X	X	X	X	X	X	X	X		X	X	X	Configuration Single Axis Interpolator
P-AXIS-00251	kenngr.	auto_call_ax	X	X	X	X	X	X	X	X		X	X	X	Configuration

														Single Axis Interpolator
P-AXIS-00252	kenngr.	consider_total_brake_dist	X	X	X	X	X	X	X		X	X	X	Configuration Single Axis Interpolator
P-AXIS-00253	kenngr.	gantry_synchronous_slave_homing			X						X	X		Gantry
P-AXIS-00254	kenngr.	cnc_controlled_stop_after_error			X						X	X		Gantry
P-AXIS-00255	vorsteuer.	default_active	X	X	X	X	X	X	X		X	X	X	Feedforward control
P-AXIS-00256	vorsteuer.	global_disable	X	X	X	X	X	X	X		X	X	X	Feedforward control
P-AXIS-00257	kenngr.	probing_signal_via_plc	X	X	X	X	X	X	X		X	X		Measurement
P-AXIS-00258	kenngr.	tracking_offset_remain	X	X	X	X	X	X	X		X	X	X	Tracking
P-AXIS-00259	handbetrieb.hb.	a_feedh	X	X	X	X	X	X	X		X	X		Manual mode
P-AXIS-00260	antr.profibus.	velocity_command_control	X	X		X	X	X			X	X	X	Position controller, Providrive
P-AXIS-00261	kenngr.	gantry_deskew_trigger	X	X	X	X	X	X	X		X	X		Gantry
P-AXIS-00262	kenngr.	coll_moving_dir_inverted	X	X	X	X	X	X	X		X			Collision detection
P-AXIS-00263	kenngr.	coll_zero_position_offset	X	X	X	X	X	X	X		X			Collision detection
P-AXIS-00264	antr.sercos.	op_mode_for_velocity_control			X								X	Switching of operation mode for spindle
P-AXIS-00265	antr.	velocity_position_control_on			X								X	Switching of operation mode for spindle
P-AXIS-00266	antr.	velocity_position_control_off			X								X	Switching of operation mode

														for spindle
P-AXIS-00267	kenngr.	coll_use_a_emergency	X	X	X	X	X	X	X	X		X		Collision detection
P-AXIS-00268	getriebe[i].	vb_not_referenced	X	X	X	X	X	X	X	X		X	X	Homing
P-AXIS-00269	kenngr.	meas_signal_drive	X	X	X	X	X	X	X	X		X	X	Measurement
P-AXIS-00270	getriebe[i].	slope_type	X	X	X	X	X	X	X	X		X	X	Dynamic value
P-AXIS-00271	lr_param.	temp_comp	X	X	X	X	X	X	X	X		X	X	Compensation
P-AXIS-00272	lr_param.	temp_comp_position_0	X	X	X	X	X	X	X	X		X	X	Compensation
P-AXIS-00273	lr_param.	temp_comp_offset_0	X	X	X	X	X	X	X	X		X	X	Compensation
P-AXIS-00274	lr_param.	temp_comp_coefficient	X	X	X	X	X	X	X	X		X	X	Compensation
P-AXIS-00275	lr_param.	temp_comp_n_cycles	X	X	X	X	X	X	X	X		X	X	Compensation
P-AXIS-00276	lr_param.	field_bus_allows_optimized_schedule	X	X	X	X	X	X	X	X		X	X	Position controler
P-AXIS-00277	kenngr.	prog_move_requires_homing	X	X		X	X	X				X	X	Homing
P-AXIS-00278	kenngr.	set_refpos_mode	X	X		X	X	X				X	X	Homing
P-AXIS-00279	kenngr.	set_refpos_offset	X	X		X	X	X				X	X	Homing
P-AXIS-00280	getriebe[i].Islope_profile.	d_grenz_stufe_1	X	X	X	X	X	X	X	X				X Dynamic value for lin. slope (only SAI)
P-AXIS-00281	getriebe[i].Islope_profile.	d_grenz_stufe_2	X	X	X	X	X	X	X	X				X Dynamic value for lin. slope (only SAI)
P-AXIS-00282	getriebe[i].Islope_profile.	d_stufe_1	X	X	X	X	X	X	X	X				X Dynamic value for lin. slope (only SAI)
P-AXIS-00283	getriebe[i].Islope_profile.	d_stufe_2	X	X	X	X	X	X	X	X				X Dynamic value for lin. slope (only SAI)
P-AXIS-00284	kenngr.	gantry_max_homing_dist	X	X		X	X	X				X	X	Gantry
P-AXIS-00285	getriebe[i].	a_ref	X	X		X	X	X				X	X	Homing

P-AXIS-00286	getriebe[i].	tr_ref	X	X		X	X	X			X	X		Homing
P-AXIS-00287	kenngr.	disable_super_imposed	X	X	X	X	X	X	X		X	X	X	Configuration Single Axis Interpolator
P-AXIS-00288	cam_gear.	is_master	X	X	X	X	X	X	X		X	X	X	Configuration Single Axis Interpolator
P-AXIS-00289	cam_gear.	v_diff_percent	X	X	X	X	X	X	X		X	X	X	Configuration Single Axis Interpolator
P-AXIS-00290	cam_gear.	time_in_window	X	X	X	X	X	X	X		X	X	X	Configuration Single Axis Interpolator
P-AXIS-00291	cam_gear.	time_out_in_window	X	X	X	X	X	X	X		X	X	X	Configuration Single Axis Interpolator
P-AXIS-00292	getriebe[i].dynamik.	a_w_max	X	X	X	X	X	X	X		X	X		Dynamic value for lin. + nonlin. slope
P-AXIS-00293	getriebe[i].dynamik.	tr_w_min	X	X	X	X	X	X	X		X	X		Dynamic value for lin. + nonlin. slope
P-AXIS-00294	kenngr.	homing_overflow_evaluation					X				X	X	X	Homing
P-AXIS-00295	antr.canopen.	probing_input_number							X		X	X	X	Measurement
P-AXIS-00296	lr_hw[i].	encoder_range							X		X	X	X	Overflow handling Encoder
P-AXIS-00297	kopf.	log_achs_name	X	X	X	X	X	X	X		X	X	X	Default axis name
P-AXIS-00298	kenngr.	max_reference_position_offset			X				X		X	X	X	Homing

P-AXIS-00299	kenngr. or kenngr.homing.	homing_type								X		X	X	X	Homing
P-AXIS-00300	cam_gear.	mv_type	X	X	X	X	X	X	X	X		X	X	X	Configuration Single Axis Interpolator
P-AXIS-00301	cam_gear.	mv_nbr_cycles	X	X	X	X	X	X	X	X		X	X	X	Configuration Single Axis Interpolator
P-AXIS-00302	cam_gear.	delay_time	X	X	X	X	X	X	X	X		X	X	X	Configuration Single Axis Interpolator
P-AXIS-00303	cam_gear.	v_max_slave	X	X	X	X	X	X	X	X		X	X	X	Configuration Single Axis Interpolator
P-AXIS-00304	cam_gear.	a_max_slave	X	X	X	X	X	X	X	X		X	X	X	Configuration Single Axis Interpolator
P-AXIS-00305	cam_gear.	v_phasing	X	X	X	X	X	X	X	X		X	X	X	Configuration Single Axis Interpolator
P-AXIS-00306	cam_gear.	a_phasing	X	X	X	X	X	X	X	X		X	X	X	Configuration Single Axis Interpolator
P-AXIS-00307	cam_gear.	d_phasing	X	X	X	X	X	X	X	X		X	X	X	Configuration Single Axis Interpolator
P-AXIS-00308	cam_gear.	keep_coupling_on_lr_error	X	X	X	X	X	X	X	X		X	X	X	Configuration Single Axis Interpolator
P-AXIS-00309	cam_gear.	keep_coupling_on_fe_drop	X	X	X	X	X	X	X	X		X	X	X	Configuration Single Axis Interpolator

P-AXIS-00310	getriebe[i].	allow_resolution_above_on e												Position controler
P-AXIS-00311	getriebe[i].	vb_monitor	X	X	X	X	X	X	X		X	X	X	Velocity monitoring
P-AXIS-00312	kenngr.	enable_speed_monitoring	X	X	X	X	X	X	X		X	X	X	Velocity monitoring
P-AXIS-00313	kenngr.	enable_feed_enable	X	X	X	X	X	X	X		X	X	X	Velocity monitoring
P-AXIS-00314	getriebe[i].	vb_torq_limit_max			X						X	X	X	Velocity monitoring
P-AXIS-00315	antr.profibus.	read_abs_pos_from_drive						X			X	X	X	Homing, Profidrive
P-AXIS-00316	antr.profibus.	p1042						X			X	X	X	Homing, Profidrive
P-AXIS-00317	antr.profibus.	p1043						X			X	X	X	Homing, Profidrive
P-AXIS-00318	antr.profibus.	read_abs_pos_mode						X			X	X	X	Homing, Profidrive
P-AXIS-00319	filter[i].	enable	X	X	X	X	X	X	X		X			Position setpoint filter
P-AXIS-00320	antr.	operation_mode			X						X	X	X	Definition of operation mode
P-AXIS-00321	antr.	reference_cam_signal			X		X		X		X	X	X	Homing
P-AXIS-00322	kenngr.	in_add_interface_pos_as_of fset	X	X	X	X	X	X	X		X	X	X	Position controler
P-AXIS-00323	lr_param.	incr_quant	X	X	X	X	X	X	X		X	X	X	Position controler
P-AXIS-00324	getriebe[i].	torque_offset			X		X		X		X	X	X	Torque configuration
P-AXIS-00325	antr.	torque_scale_num			X		X		X		X	X	X	Torque configuration
P-AXIS-00326	antr.	torque_scale_denom			X	X	X	X	X		X	X		Torque configuration

P-AXIS-00327	kenngr.	pos_corr_drive_enable			X			X	X		X	X	X	Enabling of drive controller
P-AXIS-00328	lr_param.	distance_control_on			X			X	X		X	X		Distance controlled spindle
P-AXIS-00329	kenngr.	ref_cam_is_limit_switch	X	X	X	X	X	X	X		X	X	X	Homing
P-AXIS-00330	kenngr.	meas_signal_fixed_stop		X	X	X	X	X	X		X	X		Measurement on a fixed stop
P-AXIS-00331	kenngr.	fixed_stop_pos_lag_limit		X	X	X	X	X	X		X	X		Measurement on a fixed stop
P-AXIS-00332	kenngr.	fixed_stop_nbr_cycles		X	X	X	X	X	X		X	X		Measurement on a fixed stop
P-AXIS-00333	getriebe[i].homing.	torq_move_velocity			X					X	X	X	X	Homing on a fixed stop
P-AXIS-00334	getriebe[i].homing.	torq_move_acceleration			X					X	X	X	X	Homing on a fixed stop
P-AXIS-00335	getriebe[i].homing.	torq_move_jerk			X					X	X	X	X	Homing on a fixed stop
P-AXIS-00336	antr.profibus.	abs_pos_revolutions						X			X	X	X	Profidrive
P-AXIS-00337	vorsteuer.	jerk_fact_num	X	X	X	X	X	X	X		X	X	X	Feedforward control
P-AXIS-00338	vorsteuer.	jerk_fact_denom	X	X	X	X	X	X	X		X	X	X	Feedforward control
P-AXIS-00339	getriebe[i].dynamik.	j_trans_c0	X	X	X	X	X	X	X		X	X		Jerk limit value for lin. + nonlin. slope
P-AXIS-00340	getriebe[i].dynamik.	j_trans_c1	X	X	X	X	X	X	X		X	X		Jerk limit value for lin. + nonlin. slope
P-AXIS-00341	antr.profibus.	read_abs_pos_offset						X			X			Profidrive
P-AXIS-00342	getriebe[i].homing.	torq_move_torque_limit			X					X	X	X	X	Homing on a fixed stop

P-AXIS-00343	getriebe[i].homing.	torq_detect_torque_limit			X					X	X	X	X	Homing on a fixed stop
P-AXIS-00344	kenngr.homing.	torq_min_distance			X					X	X	X	X	Homing on a fixed stop
P-AXIS-00345	kenngr.homing.	torq_max_distance			X					X	X	X	X	Homing on a fixed stop
P-AXIS-00346	kenngr.homing.	torq_homing_dir			X					X	X	X	X	Homing on a fixed stop
P-AXIS-00347	kenngr.homing.	torq_detect_velocity_limit			X					X	X	X	X	Homing on a fixed stop
P-AXIS-00348	kenngr.homing.	torq_retraction_distance			X					X	X	X	X	Homing on a fixed stop
P-AXIS-00349	kenngr.homing.	torq_homing_position			X					X	X	X	X	Homing on a fixed stop
P-AXIS-00350	kenngr.homing.	torq_detect_time			X					X	X	X	X	Homing on a fixed stop
P-AXIS-00351	kenngr.	filter_position_window	X	X	X	X	X	X	X		X	X	X	Position setpoint filter
P-AXIS-00352	antr.profibus.	disable_auto_tracking						X			X	X	X	Profidrive
P-AXIS-00353	antr.	edge_bending_input_nbr						X			X	X		Edge bending
P-AXIS-00354	antr.	encoder_overflow_offset			X		X		X		X	X	X	Homing
P-AXIS-00355	antr.	encoder_bit_range			X		X		X		X	X	X	Homing
P-AXIS-00356	kenngr.	pos_corr_ignore_feedhold			X		X				X	X	X	Enabling of drive controller
P-AXIS-00357	filter.	time_constant	X	X	X	X	X	X	X		X	X	X	Pos. setpoint filter (from V262 also for spdl.)
P-AXIS-00358	antr.	ignore_unknown_telegram_elements			X	X	X	X	X		X	X	X	Configuration drive telegram
P-AXIS-00359	handbetrieb.hb.	tr	X	X	X	X	X	X	X		X	X		Manual mode
P-AXIS-00360	handbetrieb.hb.	tr_feedh	X	X	X	X	X	X	X		X	X		Manual mode
P-AXIS-00361	vorsteuer.	timeconst_sym_filter						X			X	X	X	Pre control,

														Profidrive	
P-AXIS-00362	antr.	feed_const_num							X			X	X	X	Position controller
P-AXIS-00363	antr.	feed_const_denom							X			X	X	X	Position controller
P-AXIS-00364	antr.canopen.	zero_pulse_input_number								X		X	X	X	Homing
P-AXIS-00365	handbetrieb.ipc.	hb_v_max_track	X	X	X	X	X	X	X	X		X	X	X	Manual mode, position correction
P-AXIS-00366	handbetrieb.ipc.	hb_a_max_track	X	X	X	X	X	X	X	X		X	X	X	Manual mode, position correction
P-AXIS-00367	antr.	plc_watchdog_disable_drive_delay_time	X	X	X	X	X	X	X	X		X	X	X	Drive release at watchdog error
P-AXIS-00368	trafo.actual_pos.	enable	X	X	X	X	X	X	X	X	X	X	X	X	Axis specific actual value transformation
P-AXIS-00369	trafo.actual_pos.	id	X	X	X	X	X	X	X	X	X	X	X	X	Axis specific actual value transformation
P-AXIS-00370	trafo.actual_pos.	param[i]	X	X	X	X	X	X	X	X	X	X	X	X	Axis specific actual value transformation
P-AXIS-00371	trafo.actual_pos.input[i].	nr	X	X	X	X	X	X	X	X	X	X	X	X	Axis specific actual value transformation
P-AXIS-00372	antr.	ethercat_master_name								X		X	X	X	EtherCAT master under INtime
P-AXIS-00373	antr.dse.	brake_open_delay_time									X	X	X	X	DSE break control
P-AXIS-00374	antr.dse.	brake_open_time									X	X	X	X	DSE break

														control	
P-AXIS-00375	antr.dse.	brake_close_premature_time									X	X	X	X	DSE break control
P-AXIS-00376	cam_gear.	j_phasing	X	X	X	X	X	X	X			X	X	X	Configuration Single Axis Interpolator
P-AXIS-00377	cam_gear.	j_vel_sync	X	X	X	X	X	X	X			X	X	X	Configuration Single Axis Interpolator
P-AXIS-00378	kenngr.	keep_tp_over_reset									X	X	X	X	DSE and Measurement
P-AXIS-00379	antr.profibus.	drive_velocity_base_value							X			X	X	X	Profidrive, velocity scaling factor
P-AXIS-00380	trafo.command_pos.	enable	X	X	X	X	X	X	X	X			X		Axis specific command value transformation
P-AXIS-00381	trafo.command_pos.	id	X	X	X	X	X	X	X	X			X		Axis specific command value transformation
P-AXIS-00382	trafo.command_pos.	param[i]	X	X	X	X	X	X	X	X			X		Axis specific command value transformation
P-AXIS-00383	kenngr.multi_link[i].	lead_axis	X	X	X	X	X	X	X	X	X	X	X		Axis specific gear coupling
P-AXIS-00384	kenngr.multi_link[i].	factor_numerator	X	X	X	X	X	X	X	X	X	X	X		Axis specific gear coupling
P-AXIS-00385	kenngr.multi_link[i].	factor_denominator	X	X	X	X	X	X	X	X	X	X	X		Axis specific gear coupling
P-AXIS-00386	antr.sercos.	drive_supports_cnc_homing				X						X	X	X	SERCOS Homing
P-AXIS-00387	antr.sercos.	cnc_homing_rt_bit_layout				X						X	X	X	SERCOS Homing

P-AXIS-00388	antr.	cnc_homing_encoder			X						X	X	X	Homing
P-AXIS-00389	vorsteuer.	velocity_delay_time			X						X	X	X	Feedforward control
P-AXIS-00390	vorsteuer.	acceleration_delay_time			X						X	X	X	Feedforward control
P-AXIS-00391	getriebe[i].	load			X						X	X	X	Feedforward control
P-AXIS-00392	antr.	acc_reference_value			X						X	X	X	Feedforward control
P-AXIS-00393	kenngr.	gantry_slave_relative_homing		X	X	X	X	X	X	X	X	X	X	Homing
P-AXIS-00394	getriebe[i].dynamik.	a_overload_max	X	X	X	X	X	X	X	X	X	X		Dynamic limit values
P-AXIS-00395	getriebe[i].	pos_control_deadband	X	X	X	X	X	X		X	X	X		Position control hysteresis
P-AXIS-00396	antr.function[i].	id			X			X			X	X	X	Setting of drive functions
P-AXIS-00397	antr.function[i].	commu			X			X			X	X	X	Setting of drive functions
P-AXIS-00398	antr.function[i].	wr_ident[j]			X			X			X	X	X	Setting of drive functions
P-AXIS-00399	antr.function[i].	data_type			X			X			X	X	X	Setting of drive functions
P-AXIS-00400	antr.function[i].	startup_value			X			X			X	X	X	Setting of drive functions
P-AXIS-00401	antr.function[i].	scaling_type			X			X			X	X	X	Setting of drive functions
P-AXIS-00402	kenngr. -Spezial MEWAG	auto_restore_link_axis_position			X							X	X	Restoring of axes positions
P-AXIS-00403	antr.	abs_pos_offset	X		X		X	X	X		X	X	X	Position correction
P-AXIS-00404	lr_hw[i].	zero_pulse_search_max_distance	X	X	X	X	X	X	X		X	X	X	Homing

P-AXIS-00405	antr.	encoder_coarsening_sector			X						X	X	X	Drive encoder scaling
P-AXIS-00406	antr.	permissible_telegram_failures			X				X		X	X	X	Setting of Ethercat
P-AXIS-00407	lr_param.	dyn_monitoring_v_err	X	X	X	X	X	X	X	X	X	X	X	Dynamic monitoring position controller
P-AXIS-00408	antr.function[i]	min_limit			X			X			X	X	X	Setting of drive functions
P-AXIS-00409	antr.function[i]	max_limit			X			X			X	X	X	Setting of drive functions
P-AXIS-00410	antr.	dc_break_control_bit_nr	X					X					X	Control of DC brake
P-AXIS-00411	antr.sercos.	evaluate_drive_follows_command			X						X	X	X	Display of state on HLI
P-AXIS-00412	kenngr.	homing_max_movement_distance	X	X	X	X	X	X	X	X		X	X	Homing
P-AXIS-00413	kenngr.distc.	n_cycles			X			X	X		X	X		Distance control
P-AXIS-00414	kenngr.distc.	max_deviation (alt: max_abweichung)			X			X	X		X	X		Distance control
P-AXIS-00415	kenngr.distc.	v_max			X			X	X		X	X		Distance control
P-AXIS-00416	kenngr.distc.	a_max			X			X	X		X	X		Distance control
P-AXIS-00417	kenngr.distc.	max_act_value_change (alt: max_istw_sprung)			X			X	X		X	X		Distance control
P-AXIS-00418	kenngr.distc.	ref_offset			X			X	X		X	X		Distance control
P-AXIS-00419	kenngr.distc.	max_pos			X			X	X		X	X		Distance control
P-AXIS-00420	kenngr.distc.	min_pos			X			X	X		X	X		Distance control
P-AXIS-00421	kenngr.distc.	tolerance (alt: toleranz)			X			X	X		X	X		Distance control
P-AXIS-00422	lr_hw[i].	encoder_resolution_num	X	X	X	X	X	X	X	X	X	X	X	Position controller
P-AXIS-00423	lr_hw[i].	encoder_resolution_denom	X	X	X	X	X	X	X	X	X	X	X	Position

														controller
P-AXIS-00424	lr_hw[i].	mode_act_pos	X	X	X	X	X	X	X	X	X	X	X	Position controller
P-AXIS-00425	antr.reference_check.	bit_nr			X						X	X	X	Homing
P-AXIS-00426	antr.reference_check.	element_name			X						X	X	X	Homing
P-AXIS-00427	kenngr.	cax_output_with_generated_nc_block	X	X	X	X	X	X	X	X	X	X	X	Positioning of coupled axes
P-AXIS-00428	kenngr.distc.	check_sw_limit_switch			X			X	X		X	X		Distance control
P-AXIS-00429	antr.function[i].	mask			X			X			X	X	X	Setting of drive functions
P-AXIS-00430	antr.	probing_input_nbr			X			X	X		X	X	X	Festlegen des Messeingangs
P-AXIS-00431	dynamic_weighting.	enable	X	X	X	X	X	X	X	X	X	X		Path dependent dynamic weighting
P-AXIS-00432	dynamic_weighting.param[i].	path_limit	X	X	X	X	X	X	X	X	X	X		Path dependent dynamic weighting
P-AXIS-00433	dynamic_weighting.param[i].	velocity_fact	X	X	X	X	X	X	X	X	X	X		Path dependent dynamic weighting
P-AXIS-00434	dynamic_weighting.param[i].	acceleration_fact	X	X	X	X	X	X	X	X	X	X		Path dependent dynamic weighting
P-AXIS-00435	dynamic_weighting.param[i].	ramp_time_fact	X	X	X	X	X	X	X	X	X	X		Path dependent dynamic weighting
P-AXIS-00436	kenngr.	recalc_link_main_input_position	X	X	X	X	X	X	X	X	X	X		Gear coupling with forward mapping
P-AXIS-00437	cam_gear.	fact_a_max_correction	X	X	X	X	X	X	X	X	X	X	X	Configuration Single Axis

														Interpolator
P-AXIS-00438	kenngr.	in_add_interface_filter_time	X	X	X	X	X	X	X	X	X	X	X	Time const. filter for add. cmd. pos interf.
P-AXIS-00439	kenngr.	dyn_monitoring_v_warn	X	X	X	X	X	X	X	X	X	X	X	Dynamic monitoring
P-AXIS-00440	kenngr.	dyn_monitoring_v_err	X	X	X	X	X	X	X	X	X	X	X	Dynamic monitoring
P-AXIS-00441	kenngr.	dyn_monitoring_a_warn	X	X	X	X	X	X	X	X	X	X	X	Dynamic monitoring
P-AXIS-00442	kenngr.	dyn_monitoring_a_err	X	X	X	X	X	X	X	X	X	X	X	Dynamic monitoring
P-AXIS-00443	kenngr.	dyn_monitoring_j_warn	X	X	X	X	X	X	X	X	X	X	X	Dynamic monitoring
P-AXIS-00444	kenngr.	dyn_monitoring_j_err	X	X	X	X	X	X	X	X	X	X	X	Dynamic monitoring
P-AXIS-00445	kenngr.	gantry_independent_set_refpos	X	X	X	X	X	X	X	X	X	X	X	Setting homing pos. gantry slave axes
P-AXIS-00446	handbetrieb.default.	after_g200_g201	X	X	X	X	X	X	X	X	X	X		Manual mode
P-AXIS-00447	antr.profibus.encoder[i].	read_abs_pos_from_drive						X			X	X	X	Referencing, Profidrive, additive encoder
P-AXIS-00448	antr.profibus.encoder[i].	read_abs_pos_mode						X			X	X	X	Referencing, Profidrive, additive encoder
P-AXIS-00449	antr.profibus.encoder[i].	abs_position_offset						X			X	X	X	Referencing, Profidrive, additive encoder
P-AXIS-00450	antr.profibus.encoder[i].	shift_abs_pos						X			X	X	X	Referencing, Profidrive, additive encoder

P-AXIS-00451	antr.profibus.encoder[i].	shift_xist1						X			X	X	X	Referencing, Profidrive, additive encoder
P-AXIS-00452	antr.profibus.encoder[i].	abs_pos_offset_crc						X			X	X	X	Referencing, Profidrive, additive encoder
P-AXIS-00453	antr.profibus.	read_abs_pos_offset_crc												
P-AXIS-00454	lr_param.	allow_comp_movement_while_feedhold												
P-AXIS-00455	kenngr.	no_stop_by_channel_reset												
P-AXIS-00456	antr.canopen.	f_probe_status_inverse_edge												
P-AXIS-00457	kenngr.	enable_single_axis												PLC open command for path axes
P-AXIS-00458	kenngr.	set_in_pos_at_pos_corr												
P-AXIS-00459	antr.	add_movement_range	X		X	X	X		X		X	X	X	Überlaufbehandlung Absolutencoder
P-AXIS-00460	kenngr.	recalc_input_position_absolute	X	X	X	X	X	X	X	X	X	X	X	Gear coupling with forward mapping
P-AXIS-00461	kenngr.	activate_coupled_axes_homing_check	X	X	X	X	X	X	X	X	X	X	X	Gear coupling with forward mapping
P-AXIS-00462	antr.profibus.	suppress_life_sign_warning						X			X	X	X	Monitoring, Profidrive
P-AXIS-00463	antr.canopen.	cyclic_position_op_mode							X		X	X	X	Default mode after start-up
P-AXIS-00464	antr.canopen.	cyclic_velocity_op_mode							X		X	X	X	Default mode after start-up

P-AXIS-00465	lr_param.	prog_movement_requires_c ompensations	X	X	X	X	X	X	X	X	X	X	Monitoring of axis compensations
P-AXIS-00466	antr.	use_drive_following_error			X				X		X	X	Drive based position lag calculation
P-AXIS-00467	kenngr.	probing_offset	X	X	X	X	X	X	X	X	X	X	Measurement

参考文献

- [1]李斌, 李曦, 数控技术, 华中科技大学出版社, 2010。
- [2]吕斌杰, 孙智俊, 赵汶, 数控加工中心, 化学工业出版社, 2009.
- [3]王侃夫, 数控机床控制技术与系统, 机械工业出版社, 2007.
- [4]孙德茂, 数控磨床培训教程, 机械工业出版社, 2010.
- [5]Michael Fitzpatrick, CNC 技术, 科学出版社, 2005.
- [6]ISG, program manual